



Business Message Standard (BMS) Price Synchronisation

BMS Release: 3.1.3, SMG Name: GDS

Issue 1.3.5, 23-May-2017



Document Summary

| Document Item | Current Value |
|----------------------|---|
| Document Title | Business Message Standard (BMS) – Price Synchronisation |
| BMS Release | 3.1.3 |
| Document Version | Issue 1.3.5, 30-Jan-2017 |
| Work Group Name | GDSN |
| BMS Template Version | 2.3 |

Change Request Reference

| CR Submitter(s): | Refer to Change Request (CR) Number(s): |
|--------------------------------------|---|
| Tom Heist for RDD Team | 05-000250 |
| GS1 Australia | 08-000199 |
| GS1 France | 07-000414 |
| Olivier Mouton, Carrefour | 07-000329 |
| Sara Novak, SA2 | 07-000386 |
| Olivier Mouton, Carrefour | 07-000414 |
| Leon Plaksin, GS1 Australia | 10-000065 |
| <u>Julien Matsis GS1 New Zealand</u> | <u>14-000147</u> |
| <u>Steve Robba, 1Worldsync.</u> | <u>15-000061</u> |
| <u>Steve Robba, 1WorldSync</u> | <u>16-000327</u> |
| <u>Steve Robba, 1WorldSync</u> | <u>16-000390</u> |

Business Requirements Document (BRAD) Reference

| BRAD Title: | BRD Date: | BRAD Version |
|--|--------------|--------------|
| BRAD Price Synchronisation in the GDSN | 30-Apr-2007 | 0.0.6 |
| BRAD For GDSN Price Sync Maintenance Release 1 | 11-Dec-2007 | 1.0.0 |
| BRAD For GDSN Maintenance Release 4 | 14-Sept-2009 | 0.0.3 |
| BRAD For GDSN Maintenance Release 5 | 16-Nov-2010 | 0.0.5 |
| BRAD Multiple Expressions of Price | 8-Dec-2010 | 0.0.2 |
| BRAD For GDSN Major Release 3.X | 5-Mar-2012 | 0.0.25 |

Document Change History

| Date of Change | Version | Changed By | Reason for Change | Summary of Change | Model Build # |
|----------------|---------|-----------------|------------------------|------------------------|---------------|
| 22-March-2012 | 1.3.0 | Mark Van Eeghem | Major Release 3 of GDS | See Summary of Changes | N/A |

| Date of Change | Version | Changed By | Reason for Change | Summary of Change | Model Build # |
|--------------------|--------------|------------------|--------------------------|-------------------------------|---------------|
| 4-July-2012 | 1.3.1 | Mark Van Eeghem | Major Release 3 of GDS | See Summary of Changes | N/A |
| 6-Dec-2012 | 1.3.2 | Eric Kauz | Major Release 3 of GDSN | See Summary of Changes | |
| 16-May-2013 | 1.3.3 | Mark Van Eeghem | Major Release 3 of GDSN | See Summary of Changes | |
| 5-Aug-2013 | 1.3.4 | Mark Van Eeghem | Clean-up for Publication | See Summary of Changes | |
| <u>23-May-2017</u> | <u>1.3.5</u> | <u>Eric Kauz</u> | <u>3.1.3 Release</u> | <u>See Summary of Changes</u> | |

Disclaimer

WHILST EVERY EFFORT HAS BEEN MADE TO ENSURE THAT THE GUIDELINES TO USE THE GS1 STANDARDS CONTAINED IN THE DOCUMENT ARE CORRECT, GS1 AND ANY OTHER PARTY INVOLVED IN THE CREATION OF THE DOCUMENT HEREBY STATE THAT THE DOCUMENT IS PROVIDED WITHOUT WARRANTY, EITHER EXPRESSED OR IMPLIED, REGARDING ANY MATTER, INCLUDING BUT NOT LIMITED TO THE OF ACCURACY, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, AND HEREBY DISCLAIM ANY AND ALL LIABILITY, DIRECT OR INDIRECT, FOR ANY DAMAGES OR LOSS RELATING TO OR RESULTING FROM THE USE OF THE DOCUMENT. THE DOCUMENT MAY BE MODIFIED, SUBJECT TO DEVELOPMENTS IN TECHNOLOGY, CHANGES TO THE STANDARDS, OR NEW LEGAL REQUIREMENTS. SEVERAL PRODUCTS AND COMPANY NAMES MENTIONED HEREIN MAY BE TRADEMARKS AND/OR REGISTERED TRADEMARKS OF THEIR RESPECTIVE COMPANIES. GS1 IS A REGISTERED TRADEMARK OF GS1 AISBL.

Table of Contents

| | |
|--|-----------|
| 1. Business Domain View | 6 |
| 1.1. Problem Statement / Business Need | 6 |
| 1.2. Objective | 6 |
| 1.3. Audience | 6 |
| 1.4. References | 7 |
| 1.5. Acknowledgements | 7 |
| 1.5.1. SMG Work Group | 7 |
| 1.5.2. Design Team Members | 10 |
| 2. Business Context | 10 |
| 3. Business Transaction View | 11 |
| 3.1. Use Case Definitions – Add Trading Relationship | 11 |
| 3.2. Add Trading Relationship | 12 |
| 3.3. Update Trading Relationship | 14 |
| 3.4. Cancel Trading Relationship | 17 |
| 3.5. Discontinue Trading Relationship | 19 |
| 3.6. Synchronise Conditions | 21 |
| 3.7. Add Condition | 21 |
| 3.8. Modify Condition | 24 |
| 3.9. Withdraw Condition | 26 |
| 3.10. Discontinue a Condition | 28 |
| 3.11. Synchronise Price Type | 29 |
| 3.12. Add Item Price Type | 30 |
| 3.13. Modify Item Price Type | 33 |
| 3.14. Withdraw Item Price Type | 36 |
| 3.15. Discontinue Item Price Type | 39 |
| 4. Information Model | 41 |
| 4.1. Class Diagrams | 41 |
| 4.1.1. Codes | 41 |
| 4.1.2. Price Synchronisation Document | 42 |
| 4.1.3. Price Synchronisation Confirmation | 44 |
| 4.1.4. Item Depiction Qualifier | 47 |
| 4.1.5. Price Synchronisation Condition | 53 |
| 4.1.6. Price Synchronisation Relationship | 57 |
| 4.2. Code Lists | 60 |
| 4.2.1. Bracket Range Qualifier Code | 60 |
| 4.2.2. Component Value Type Code | 60 |
| 4.2.3. Condition Type Code | 61 |
| 4.2.4. Effective End Date Context Code | 61 |

| | | |
|-----------|---|-----------|
| 4.2.5. | Effective Start Date Context Code | 62 |
| 4.2.6. | Performance Requirement Option Code | 62 |
| 4.2.7. | Price Action Reason Code | 63 |
| 4.2.8. | Price Document Type Code | 64 |
| 4.2.9. | Price Synchronisation Special Scenario Code | 64 |
| 4.2.10. | Price Type Code | 65 |
| 4.2.11. | Price Value Qualifier Code | 66 |
| 4.2.12. | Segment Action Code | 67 |
| 5. | Business Document Example | 67 |
| 6. | Implementation Considerations | 71 |
| 6.1.1. | Bulk Update | 71 |
| 6.1.2. | Initial Load | 71 |
| 6.1.3. | Resend | 72 |
| 6.1.4. | Reload | 72 |
| 6.1.5. | Restart | 72 |
| 6.2. | Price Sequencing Rules | 73 |
| 6.2.1. | Item Price Types | 73 |
| 6.2.2. | Summary Conditions | 74 |
| 6.3. | Communicating Multiple Catalogue Item Qualifiers | 74 |
| 6.4. | Price Commentary | 75 |
| 6.5. | Special Scenario Code | 75 |
| 6.5.1. | Resynchronisation of All Price Types for Pricing Done at the Lowest Level Consumer Unit | 76 |
| 6.6. | Target Market Information Provider | 76 |
| 7. | Appendices | 76 |
| 8. | Summary of Changes | 76 |

1. Business Domain View

1.1. Problem Statement / Business Need

Currently there is limited capability for electronically communicating accurate pricing information between trading partners using global standards that

“accommodates all the different pricing business practices and facilitates an invoice amount equal to the expected payment amount equal to the actual payment”.

Globally, pricing business practices range from simple pricing and transactional pricing to component based pricing. Component based pricing includes components such as pro-motions, allowances, charges and brackets.

This BMS will also remedy the following issues with the GS1 standards in relation to Price Synchronisation:

When communicating price, partners in some target markets have a need to communicate associated prices together, as a single business unit for example:

- the list price, exclusive taxes
- the transaction price, exclusive of special taxes and VAT

| | | Start Date | Reason | GTIN | Ref ID | Dist Method | Price Type | Value | | |
|-----------|----------------|------------|----------------|--------|--------|-------------|-------------|-------|--|-------------------|
| One Price | Single Record: | 2009-04-01 | Price Increase | GTIN A | 123 | DSD | List Price | 10 | | One Business Unit |
| | | | | | | | Txn Price 1 | 11 | | |
| | | | | | | | Txn Price 2 | 12 | | |

*where Txn Price 1 = Transaction Price with special taxes
Txn Price 2 = Transaction Price with special taxes + VAT

1.2. Objective

To supply the detail design of the (specific) business transaction needed to meet the requirements of the referenced in the BRAD for Price Multiple Expressions V 0.0.2 and in the BRAD for GDSN Major Release 3.X, building on the existing BMS for GDSN Maintenance Release 2.8.

1.3. Audience

The audience would be any participant in the global supply chain engaged in the GDSN. This would include the roles of suppliers (or sellers or data source), source data pools, recipient data pools, retailers (or buyers or data recipient) and other third parties.

1.4. References

| Reference Name | Description |
|---|--|
| BRAD Price Synchronisation in the GDSN V 0.0.5 | Requirements documentation for applying price synchronisation in the GDSN. |
| Align – BMS Trading Partner Profile | Approved global standard for price synchronization outside of the GDSN. |
| Align – BMS Condition Document and Monetary Documents | Approved global standard for price synchronization outside of the GDSN. |
| BRAD GDSN Price Sync Maintenance Release V 1.0.0 | Requirements for maintenance update of Price Synchronisation messages. |
| BRAD For GDSN Maintenance Release 4 | Maintenance Release CRs. |
| BRAD For Price Multiple Expressions | Requirements documentation for dealing with multiple price. |
| BRAD For GDSN Major Release 3 | Major Release CRs |
| BMS Shared Common Components Release 3 | Documents the data elements that are common in use between GDSN and eCom. |

1.5. Acknowledgements

The following is a list of individuals (and their companies) who participated in the creation, review and approval of this BMS.

1.5.1. SMG Work Group

| Function | Name | Company / organisation |
|-----------------|-------------------|---------------------------------|
| SMG Chair | Scott Brown | GS1 US |
| SMG Co-Chair | Robin Kidd | Nestle |
| SMG Co-Chair | Steve Robba | SA2 |
| Process Manager | Justin Childs | GS1 Global Office |
| SMG Member | Vanessa Frosch | 1SYNC |
| SMG Member | Steve Vazzano | 1SYNC |
| SMG Member | Donna Yeksigian | 1SYNC |
| SMG Member | Rita Joyce | 3M Company |
| SMG Member | Jon Peterson | 3M Company |
| SMG Member | Cynthia Poetker | Abbott Laboratories Inc. |
| SMG Member | Marcel Yska | Ahold (Netherlands) |
| SMG Member | Mickey Atkins | Ahold (USA) |
| SMG Member | Frank Heemelaar | Albert Heijn |
| SMG Member | Armand Schins | Albert Heijn |
| SMG Member | Tom Eric Schmidt | August Storck KG |
| SMG Member | Alasdair Garbett | Autogrill Retail UK Ltd t/a WDF |
| SMG Member | Sara Halfmann | Best Buy Co., Inc. |
| SMG Member | Bekki Windsperger | Best Buy Co., Inc. |

| Function | Name | Company / organisation |
|------------|------------------------|------------------------------------|
| SMG Member | Maureen Wissel | Best Buy Co., Inc. |
| SMG Member | Ed Jesus | Chep |
| SMG Member | James Sykes | Chep |
| SMG Member | Alison Bartlet | Commport Communications Int'l Inc. |
| SMG Member | Nadine Radomski | Dean Foods Company |
| SMG Member | Norbert Roehl | Edeka Zentrale AG & Co. KG |
| SMG Member | Patrick Roy | FSE, Inc. |
| SMG Member | Rajiv Singh | Garud Technology Services Inc |
| SMG Member | Carol Edison | General Mills, Inc. |
| SMG Member | Joy Schneck | General Mills, Inc. |
| SMG Member | Ardetha Bradley | Georgia Pacific |
| SMG Member | Milan Vacval | Gladson Interactive |
| SMG Member | Mitch Fortier | GS1 Australia |
| SMG Member | Justin Middleton | GS1 Australia |
| SMG Member | Sue Schmid | GS1 Australia |
| SMG Member | Stephan Wijnker | GS1 Australia |
| SMG Member | Eugen Sehorz | GS1 Austria |
| SMG Member | Kristel Lai | GS1 Canada |
| SMG Member | Rita Laur | GS1 Canada |
| SMG Member | Reza Mahdiani | GS1 Canada |
| SMG Member | Natalia Yusseem | GS1 Canada |
| SMG Member | Giovanni Biffi | GS1 Colombia |
| SMG Member | Eddy Merrill | GS1 Community Room Staff |
| SMG Member | Mike Mowad | GS1 Community Room Staff |
| SMG Member | Pertti Hakala | GS1 Finland |
| SMG Member | Jean-Luc Leblond | GS1 France |
| SMG Member | Patricia Perrier | GS1 France |
| SMG Member | Roman Strand | GS1 Germany |
| SMG Member | Tanja Thomsen | GS1 Germany |
| SMG Member | János Gyuris | GS1 Hungary |
| SMG Member | Krisztina Vatai | GS1 Hungary |
| SMG Member | Dani Yusdiar | GS1 Indonesia |
| SMG Member | Siobhain Duggan | GS1 Ireland |
| SMG Member | Stefan Gathmann | GS1 Ireland |
| SMG Member | Brendan Kernan | GS1 Ireland |
| SMG Member | Andrea Ausili | GS1 Italy |
| SMG Member | Federico Mittersteiner | GS1 Italy |
| SMG Member | Hideki Ichihara | GS1 Japan |
| SMG Member | Carlos Ramos | GS1 Mexico |

| Function | Name | Company / organisation |
|------------|----------------------|---|
| SMG Member | Gabriel Sobrino | GS1 Netherlands |
| SMG Member | Fiona van der Linde | GS1 South Africa / Consumer Goods Council of South Africa |
| SMG Member | Xavier Pujol | GS1 Spain |
| SMG Member | Pere Rosell | GS1 Spain |
| SMG Member | Peter Jönsson | GS1 Sweden |
| SMG Member | Staffan Olsson | GS1 Sweden |
| SMG Member | Mats Wennebo | GS1 Sweden |
| SMG Member | Richard Chresta | GS1 Switzerland |
| SMG Member | Thanh Reichen | GS1 Switzerland |
| SMG Member | Neil Gray | GS1 UK |
| SMG Member | Shan Welch | GS1 UK |
| SMG Member | Rich Richardson | GS1 US |
| SMG Member | Steven Rosenberg | GS1 US |
| SMG Member | Tracey Davies | GXS (UK) |
| SMG Member | Joanna Stewart | GXS (US) |
| SMG Member | Kathrin Kiesel | Henkel AG. & Co. KGaA |
| SMG Member | Rob Hoffman | Hershey Company (The) |
| SMG Member | Christine Nye | Hershey Company (The) |
| SMG Member | Eric Ginsburg | HJ Heinz |
| SMG Member | Sylvia Rubio Alegren | ICA AB |
| SMG Member | Thomas Werthwine | Johnson & Johnson |
| SMG Member | Betty Tyson | Knouse Foods Cooperative, Inc |
| SMG Member | Leslie Henderson | Kraft Foods, Inc. |
| SMG Member | Barbara Munro | Kraft Foods, Inc. |
| SMG Member | Ryan Richard | Kraft Foods, Inc. |
| SMG Member | Jillian Wille | Kraft Foods, Inc. |
| SMG Member | Robert West | L'Oreal |
| SMG Member | Denton Clark | Lockheed Martin |
| SMG Member | Hanjoerg Lerch | METRO Group |
| SMG Member | Véra Feuerstein | Nestle |
| SMG Member | Joseph Bohning | Nestle Purina PetCare |
| SMG Member | Greg Buckley | PepsiCo, Inc. |
| SMG Member | Gina Tomassi | PepsiCo, Inc. |
| SMG Member | Noam Bronstein | Procter & Gamble Co. |
| SMG Member | Jonathan Bemrose | R&R Ice Cream |
| SMG Member | Sascha Kasper | SA2 Worldsync GmbH |
| SMG Member | Selcuk Ovuc | SA2 Worldsync GmbH |
| SMG Member | Ute Trelle | SA2 Worldsync GmbH |
| SMG Member | Felix Loecher | SAP AG |

| Function | Name | Company / organisation |
|------------|------------------|---|
| SMG Member | Maxim Stafeyev | SKB Kontur |
| SMG Member | Greg Zwanziger | SUPERVALU |
| SMG Member | John Fitzpatrick | Syncnicity for US Department of Defense |
| SMG Member | Jason Lavik | Target Corporation |
| SMG Member | Phyllis Koch | The Schwan Food Company |
| SMG Member | Werner Kolb | Unilever N.V. |
| SMG Member | Audrey Wiggins | Wal-Mart Stores, Inc. |
| SMG Member | Mac Young | Waldo County General Hospital |
| SMG Member | Kristin Andersen | Wegmans Food Markets |
| SMG Member | Jan Jaworski | Wilton Industries, Inc. |

1.5.2. Design Team Members

| Function | Name | Organisation |
|----------------------------|-----------------|--------------|
| Standards Content Lead | Mark Van Eeghem | GS1 |
| Technical Development Lead | Ewa Iwicka | GS1 |
| Peer Review | Eric Kauz | GS1 |

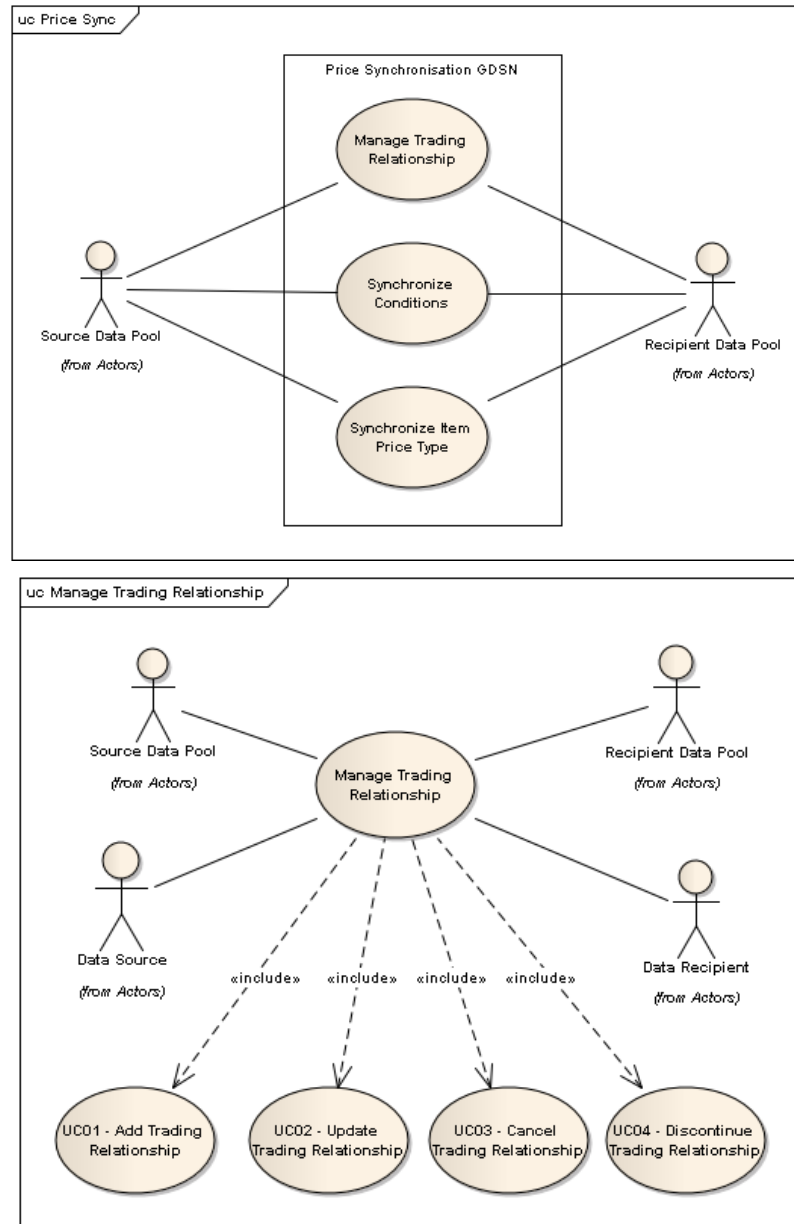
2. Business Context

| Context Category | Value(s) |
|----------------------|----------|
| Industry | All |
| Geopolitical | All |
| Product | All |
| Process | GDSN |
| System Capabilities | GS1 |
| Official Constraints | None |

3. Business Transaction View

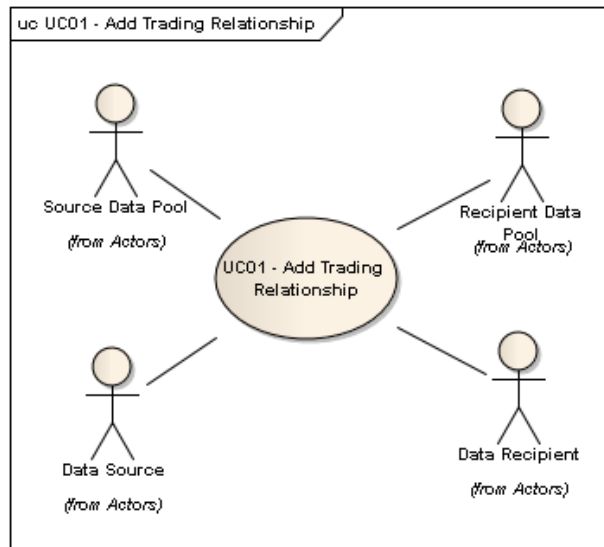
3.1. Use Case Definitions – Add Trading Relationship

Figure 3-1 Use Case Diagram: Price Synchronisation GDSN



3.2. Add Trading Relationship

Figure 3-2 Add Trading Relationship Use Case Diagram

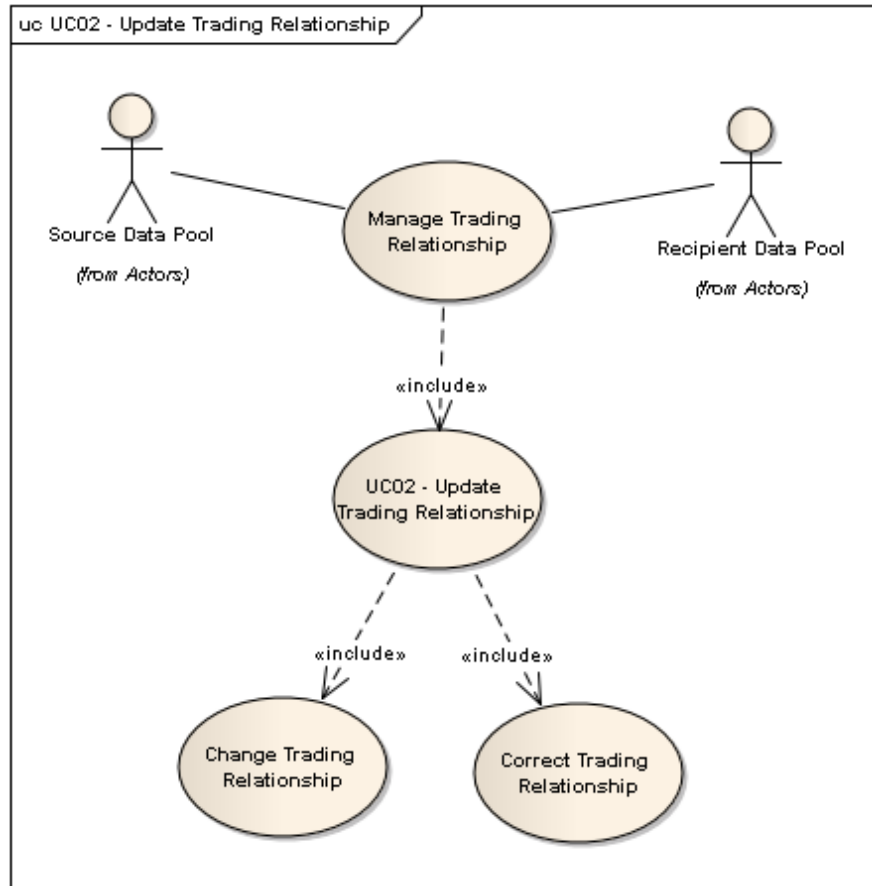


| Use Case ID | UC-1 | | | | | | | | | | | | | | | | | | |
|----------------------|---|--|-------|---------------|---|-----|-----------------------|---|-----|---|---|-----|--|---|----------------|--|---|-----|---|
| Use Case Name | Add Trading Relationship | | | | | | | | | | | | | | | | | | |
| Use Case Description | This use case establishes a price synchronisation trading partner relationship. | | | | | | | | | | | | | | | | | | |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient | | | | | | | | | | | | | | | | | | |
| Performance Goals | Initiate a price synchronisation relationship. | | | | | | | | | | | | | | | | | | |
| Preconditions | Trading partners have established a trading partner agreement including price synchronisation relationships, agreed-to pricing conditions; and are engaged in item synchronisation. | | | | | | | | | | | | | | | | | | |
| Post conditions | Price synchronisation relationship is active. | | | | | | | | | | | | | | | | | | |
| Scenario | <p>Begins when... The data source notifies their SDP of a new relationship and the SDP creates a price synchronisation list for the relationship. (Done outside of the network).</p> <p>Continues with...</p> <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Performs validations.</td></tr><tr><td>2</td><td>SDP</td><td>Creates a relationship by sending a price synchronisation message with a document command of “add” with a relationship segment action code of “add” to the RDP.</td></tr><tr><td>3</td><td>RDP</td><td>Receives price synchronisation message and sends relationship information to data recipient.</td></tr><tr><td>4</td><td>Data Recipient</td><td>Receives the trading relationship information and confirms the relationship by responding with an “RECEIVED” response. The confirmation response message is sent to the RDP.</td></tr><tr><td>5</td><td>RDP</td><td>Sends the confirmation response message to the SDP.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Performs validations. | 2 | SDP | Creates a relationship by sending a price synchronisation message with a document command of “add” with a relationship segment action code of “add” to the RDP. | 3 | RDP | Receives price synchronisation message and sends relationship information to data recipient. | 4 | Data Recipient | Receives the trading relationship information and confirms the relationship by responding with an “RECEIVED” response. The confirmation response message is sent to the RDP. | 5 | RDP | Sends the confirmation response message to the SDP. |
| Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | |
| 1 | SDP | Performs validations. | | | | | | | | | | | | | | | | | |
| 2 | SDP | Creates a relationship by sending a price synchronisation message with a document command of “add” with a relationship segment action code of “add” to the RDP. | | | | | | | | | | | | | | | | | |
| 3 | RDP | Receives price synchronisation message and sends relationship information to data recipient. | | | | | | | | | | | | | | | | | |
| 4 | Data Recipient | Receives the trading relationship information and confirms the relationship by responding with an “RECEIVED” response. The confirmation response message is sent to the RDP. | | | | | | | | | | | | | | | | | |
| 5 | RDP | Sends the confirmation response message to the SDP. | | | | | | | | | | | | | | | | | |

| | <table><tr><td>6</td><td>SDP</td><td>Updates the price synchronisation list and sends confirmation information to the data source.</td></tr></table> <p>Ends when... data source receives the confirmation response and the pricing synchronisation is active.</p> | 6 | SDP | Updates the price synchronisation list and sends confirmation information to the data source. | | | | | | |
|-----------------------------|--|---|-------|---|--|-----|---|---|----------------|--|
| 6 | SDP | Updates the price synchronisation list and sends confirmation information to the data source. | | | | | | | | |
| Alternative Scenario | <p><i>The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps</i></p> <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td></td><td>All</td><td>The scenario shows the most anticipated choreography where the SDP sends to the RDP; but the SDP may send directly to the data recipient in situations where the SDP is also the data recipient's RDP. To reduce complexity the later is not shown in the activity steps in any scenario.</td></tr><tr><td>3</td><td>Data Recipient</td><td>Data recipient responds with a confirmation status other than RECEIVED or no response is sent by the data recipient. See related rules below for status codes and their actions.</td></tr></table> | Step # | Actor | Activity Step | | All | The scenario shows the most anticipated choreography where the SDP sends to the RDP; but the SDP may send directly to the data recipient in situations where the SDP is also the data recipient's RDP. To reduce complexity the later is not shown in the activity steps in any scenario. | 3 | Data Recipient | Data recipient responds with a confirmation status other than RECEIVED or no response is sent by the data recipient. See related rules below for status codes and their actions. |
| Step # | Actor | Activity Step | | | | | | | | |
| | All | The scenario shows the most anticipated choreography where the SDP sends to the RDP; but the SDP may send directly to the data recipient in situations where the SDP is also the data recipient's RDP. To reduce complexity the later is not shown in the activity steps in any scenario. | | | | | | | | |
| 3 | Data Recipient | Data recipient responds with a confirmation status other than RECEIVED or no response is sent by the data recipient. See related rules below for status codes and their actions. | | | | | | | | |
| Related Requirements | Not Applicable | | | | | | | | | |
| Related Rules | <ol style="list-style-type: none">Confirmation status codes other than "received" are: review – message received and no action taken yet; synchronized – message received and implemented into the backend system; reject – message received and terms of a specific price message segment were rejected or the data recipient wishes to terminate the price synchronisation relationship.Action codes for the header segment other than initial load are: resend – used to indicate the message is to recover a lost or missing message; restart – used where a data recipient had rejected an item's pricing and wishes to resume synchronisation; and reload – used to "start over" by sending all current and future pricing.A price synchronisation relationship can have only one active relationship segment at a time.If the Document Header is "ADD", the Price Document ID must = "1"When relationship action document header equals "add", there are no dependency checks.The data recipient can override a previous confirmation status with another one through a confirmation response.Multiple confirmations can be sent by data recipients for a single price message or message segment. For example, a data recipient can send a status of 'Received' followed by 'Synchronised'. Exception: a data recipient cannot modify a status of Rejected. A Restart is the only way to re-initiate synchronisation on a Price Type that has been rejected.Reason code is conditional on the confirmation status being "Review". If reason code is present, ensure that confirmation status in "Review". | | | | | | | | | |

3.3. Update Trading Relationship

Figure 3-3 Update Trading Relationship Use Case Diagram



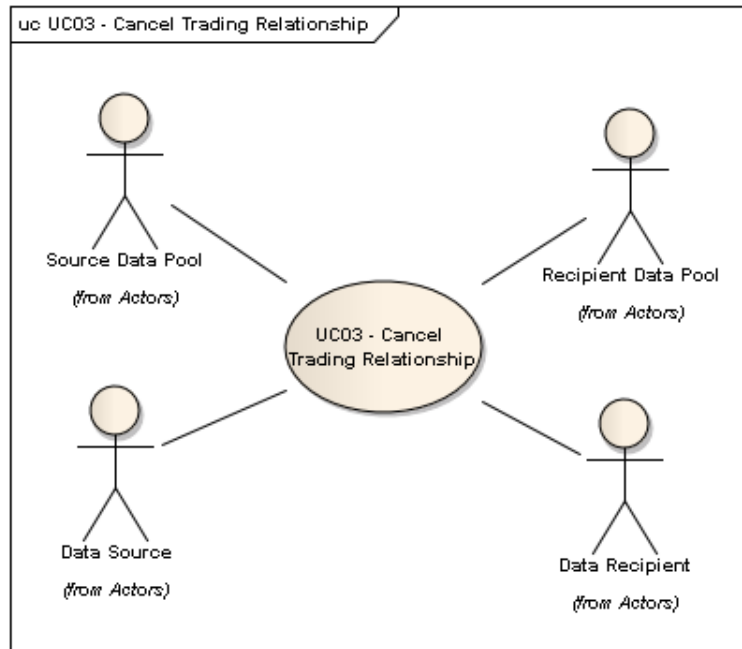
| | |
|-----------------------------|--|
| Use Case ID | UC-2 |
| Use Case Name | Update Trading Relationship |
| Use Case Description | This use case maintains the price synchronisation trading partner relationship through either modifications or corrections to the relationship data. |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient |
| Performance Goals | To update the price synchronisation relationship. |
| Preconditions | Trading partners have established a trading partner agreement including price synchronisation relationships, agreed-to pricing conditions; and are engaged in item synchronisation. Trading relationship data has been previously received and accepted by the data recipient. |
| Post conditions | Price synchronisation relationship is updated. |

| Scenario | Begins when... The data source notifies their SDP of updates to a trading relationship (done outside of the network). | | | | | | | | | | | | | | | | | | | | | |
|---|---|--|---|---------------|---|-----|--|---|----------------|---|---|-----|--|---|----------------|--|---|-----|---|---|-----|--|
| | Continues with... | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Validates trading relationship information.</td></tr><tr><td>2</td><td>SDP</td><td>Updates the relationship by sending a price synchronisation message with a document command of "CHANGE_BY_REFRESH" with a relationship section action code of "CHANGE_BY_REFRESH" (for a modification) or "Correct" (for a correct) to the RDP.</td></tr><tr><td>3</td><td>RDP</td><td>Receives price synchronisation message and sends relationship information to data recipient.</td></tr><tr><td>4</td><td>Data Recipient</td><td>Receives the trading relationship information and confirms the relationship by responding with a an "accept" "RECEIVED" response. The confirmation response message is sent to the RDP.</td></tr><tr><td>5</td><td>RDP</td><td>Sends the confirmation response message to the SDP.</td></tr><tr><td>6</td><td>SDP</td><td>Sends confirmation information to the data source.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Validates trading relationship information. | 2 | SDP | Updates the relationship by sending a price synchronisation message with a document command of "CHANGE_BY_REFRESH" with a relationship section action code of "CHANGE_BY_REFRESH" (for a modification) or "Correct" (for a correct) to the RDP. | 3 | RDP | Receives price synchronisation message and sends relationship information to data recipient. | 4 | Data Recipient | Receives the trading relationship information and confirms the relationship by responding with a an "accept" "RECEIVED" response. The confirmation response message is sent to the RDP. | 5 | RDP | Sends the confirmation response message to the SDP. | 6 | SDP | Sends confirmation information to the data source. |
| | Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | |
| | 1 | SDP | Validates trading relationship information. | | | | | | | | | | | | | | | | | | | |
| | 2 | SDP | Updates the relationship by sending a price synchronisation message with a document command of "CHANGE_BY_REFRESH" with a relationship section action code of "CHANGE_BY_REFRESH" (for a modification) or "Correct" (for a correct) to the RDP. | | | | | | | | | | | | | | | | | | | |
| | 3 | RDP | Receives price synchronisation message and sends relationship information to data recipient. | | | | | | | | | | | | | | | | | | | |
| | 4 | Data Recipient | Receives the trading relationship information and confirms the relationship by responding with a an "accept" "RECEIVED" response. The confirmation response message is sent to the RDP. | | | | | | | | | | | | | | | | | | | |
| 5 | RDP | Sends the confirmation response message to the SDP. | | | | | | | | | | | | | | | | | | | | |
| 6 | SDP | Sends confirmation information to the data source. | | | | | | | | | | | | | | | | | | | | |
| Ends when... data source receives the confirmation response. | | | | | | | | | | | | | | | | | | | | | | |
| Alternative Scenario | <i>The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps</i> | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td></td><td>All</td><td>The scenario shows the most anticipated choreography where the SDP sends to the RDP; but the SDP may send directly to the data recipient in situations where the SDP is also the data recipients RDP. To reduce complexity the later is not shown in the activity steps in any scenario.</td></tr><tr><td>3</td><td>Data Recipient</td><td>Data Recipient responds with a confirmation status other than accept "RECEIVED". See related requirements below for status codes and their actions.</td></tr></table> | Step # | Actor | Activity Step | | All | The scenario shows the most anticipated choreography where the SDP sends to the RDP; but the SDP may send directly to the data recipient in situations where the SDP is also the data recipients RDP. To reduce complexity the later is not shown in the activity steps in any scenario. | 3 | Data Recipient | Data Recipient responds with a confirmation status other than accept "RECEIVED". See related requirements below for status codes and their actions. | | | | | | | | | | | | |
| | Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | |
| | All | The scenario shows the most anticipated choreography where the SDP sends to the RDP; but the SDP may send directly to the data recipient in situations where the SDP is also the data recipients RDP. To reduce complexity the later is not shown in the activity steps in any scenario. | | | | | | | | | | | | | | | | | | | | |
| 3 | Data Recipient | Data Recipient responds with a confirmation status other than accept "RECEIVED". See related requirements below for status codes and their actions. | | | | | | | | | | | | | | | | | | | | |
| Related Requirements | Not Applicable | | | | | | | | | | | | | | | | | | | | | |

| | |
|----------------------|---|
| Related Rules | <ol style="list-style-type: none"> 1. Confirmation status codes other than accept<u>received</u> are: review – message received and no action taken yet; synchronized – message received and implemented into the backend system; reject – message received and terms of a specific price message segment were rejected or the Data Recipient wishes to terminate the price synchronisation relationship. 2. A confirmation status of “rejected” results in a data recipient initiated termination of the trading relationship. 3. If the Document Header is “CHANGE_BY_REFRESH”, the Price Document ID must be greater than “1” 4. When Relationship action equals “CHANGE_BY_REFRESH”, a positive response must be in the sync list for the Relationship segment before any adds/modifies/corrects/deletes to any other segment are sent. Note: a positive response is defined as any confirmation response other than “rejected” or “no response”. 5. Relationship Start Effective Date can only be Corrected, not modified. 6. The data recipient can override a previous confirmation status with another one through a confirmation response. 7. Multiple confirmations can be sent by data recipients for a single price message or message segment. For example, a data recipient can send a status of ‘Received’ followed by ‘Synchronised’. 8. Start Effective Date can be corrected if it is not yet in effect (future date). 9. If a revised Start Effective Date is required for a relationship that is not yet in effect, the relationship must be deleted or corrected. If a revised Start Effective Date is required for a relationship that is in effect, then you must set the End Effective Date and send in a new relationship with a new Start Effective Date. 10. Reason code is conditional on the confirmation status being “Review”. If reason code is present, ensure that confirmation status in “Review”. 11. For the Price Synchronisation Message, the Segment Action Code of “CHANGE_BY_REFRESH” assumes full refresh of the message segment only. |
|----------------------|---|

3.4. Cancel Trading Relationship

Figure 3-4 Cancel Trading Relationship Use Case Diagram

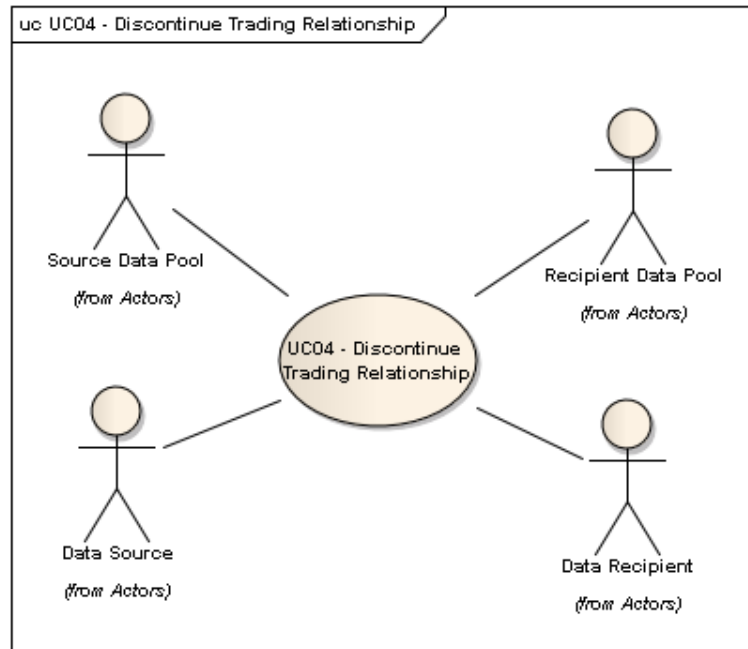


| Use Case ID | UC-3 | | | | | | | | | | | | |
|----------------------|--|--|-----------------------|---------------|---|-----|-----------------------|---|-----|--|---|-----|--|
| Use Case Name | Cancel Trading Relationship | | | | | | | | | | | | |
| Use Case Description | This use case terminates a specific price synchronisation trading partner relationship that has not yet taken effect. | | | | | | | | | | | | |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient | | | | | | | | | | | | |
| Performance Goals | To terminate a price synchronisation relationship. | | | | | | | | | | | | |
| Preconditions | Trading partners have established a trading partner agreement including price synchronisation relationships, agreed-to pricing conditions; and are engaged in item synchronisation. Trading relationship data has been previously received by the data recipient. | | | | | | | | | | | | |
| Post conditions | Price synchronisation relationship is terminated. | | | | | | | | | | | | |
| Scenario | Begins when... The data source notifies their SDP of the need to terminate a trading relationship (done outside of the network). | | | | | | | | | | | | |
| | Continues with... | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Performs validations.</td></tr><tr><td>2</td><td>SDP</td><td>Terminates the relationship by sending a price synchronisation message with a document command of "CHANGE_BY_REFRESH " with a relationship section action code of "Delete" to the RDP.</td></tr><tr><td>3</td><td>RDP</td><td>Receives price synchronisation message and sends relationship information to data recipient.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Performs validations. | 2 | SDP | Terminates the relationship by sending a price synchronisation message with a document command of "CHANGE_BY_REFRESH " with a relationship section action code of "Delete" to the RDP. | 3 | RDP | Receives price synchronisation message and sends relationship information to data recipient. |
| | Step # | Actor | Activity Step | | | | | | | | | | |
| | 1 | SDP | Performs validations. | | | | | | | | | | |
| 2 | SDP | Terminates the relationship by sending a price synchronisation message with a document command of "CHANGE_BY_REFRESH " with a relationship section action code of "Delete" to the RDP. | | | | | | | | | | | |
| 3 | RDP | Receives price synchronisation message and sends relationship information to data recipient. | | | | | | | | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |

| | | | |
|-----------------------------|---|----------------|---|
| | 4 | Data Recipient | Receives the trading relationship information and confirms the relationship by responding with a an "accept RECEIVED " response. The confirmation response message is sent to the RDP. |
| | 5 | RDP | Sends the confirmation response message to the SDP. |
| | 6 | SDP | Sends confirmation information to the data source. |
| | Ends when... data source receives the confirmation response. | | |
| Alternative Scenario | <i>The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps</i> | | |
| | Step # | Actor | Activity Step |
| | | All | The scenario shows the most anticipated choreography where the SDP sends to the RDP; but the SDP may send directly to the data recipient in situations where the SDP is also the data recipient's RDP. To reduce complexity the later is not shown in the activity steps in any scenario. |
| | 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED . See related requirements below for status codes and their actions. |
| Related Requirements | Not Applicable | | |
| Related Rules | <ol style="list-style-type: none"> Confirmation status codes other than RECEIVEDaccept are: REVIEWreview – message received and no action taken yet; synchronized – message received and implemented into the backend system; REJECTEDreject – message received and terms of a specific price message segment were rejected or the data recipient wishes to terminate the price synchronisation relationship. A confirmation status of "REJECTED" is not valid for the End Trading Relationship use case. A confirmation status of "REJECTED" implies that the data recipient initiated the termination of the trading relationship. If the Document Header is "CHANGE_BY_REFRESH", the Price Document ID must be greater than "1" The "Delete" action code implies all data associated with the Relationship ID is no longer valid only for a relationship that has not taken effect. If the relationship is already in effect, the data source must send a Modify transaction (CHANGE_BY_REFRESH) and populate the End Effective Date In order to end a relationship segment, all dependent condition and price type segments need to be deleted/end dated before a delete/end date can be sent for the Relationship Segment. Can only end at a Relationship Segment ID level (i.e. if you have 3 relationship IDs identified for a trading relationship, in order to end the ENTIRE relationship, all 3 relationship IDs must be deleted/end dated). Reason code is conditional on the confirmation status being "REVIEWreview". If reason code is present, ensure that confirmation status in "REVIEWreview". For the Price Synchronisation Message, the Segment Action Code of "CHANGE_BY_REFRESH" assumes full refresh of the message segment only. | | |

3.5. Discontinue Trading Relationship

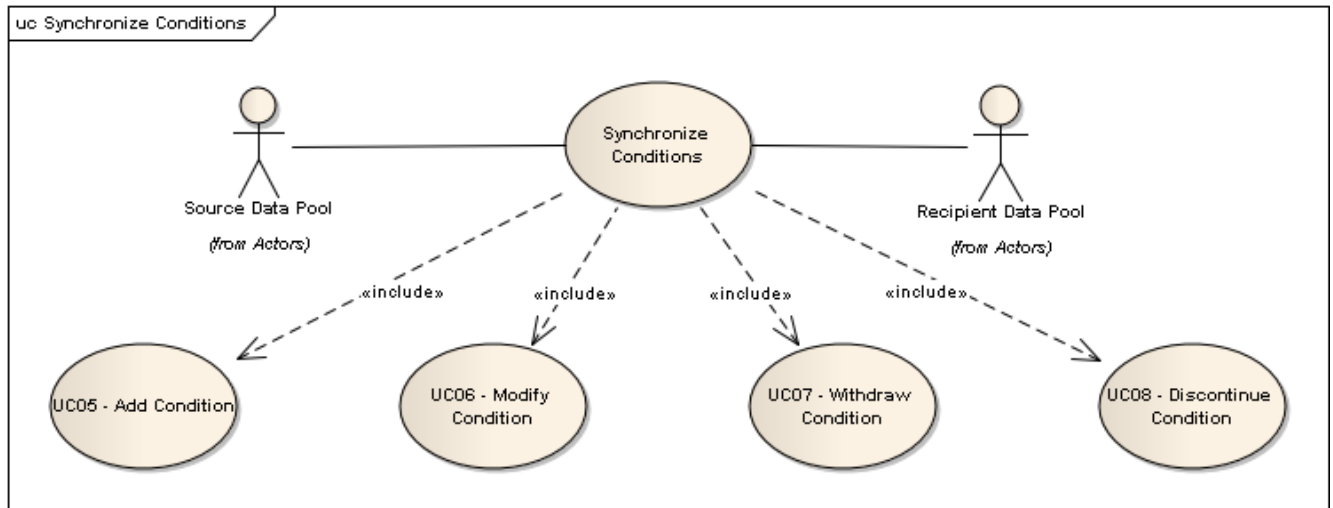
Figure 3-5 Discontinue Trading Relationship Use Case Diagram



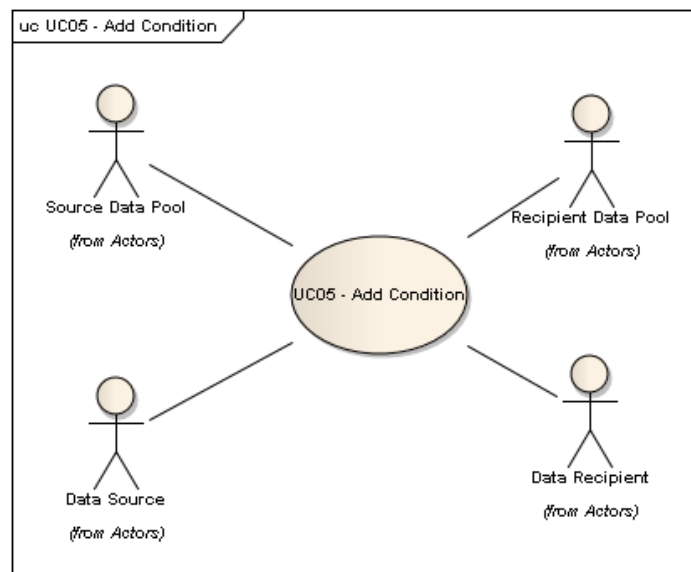
| Use Case ID | UC-4 | | | | | | | | | |
|----------------------|---|--|-------|---------------|---|-----|-----------------------|---|-----|--|
| Use Case Name | Discontinue Trading Relationship | | | | | | | | | |
| Use Case Description | This use case terminates a specific price synchronisation trading partner relationship that is currently in effect. | | | | | | | | | |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient | | | | | | | | | |
| Performance Goals | To terminate a price synchronisation relationship. | | | | | | | | | |
| Preconditions | Trading partners have established a trading partner agreement including price synchronisation relationships, agreed-to pricing conditions; and are engaged in item synchronisation. Trading relationship data has been previously received by the data recipient. The current date is greater than or equal to the effective date of the relationship. | | | | | | | | | |
| Post conditions | Price synchronisation relationship is discontinued. | | | | | | | | | |
| Scenario | <p>Begins when... The data source notifies their SDP of the need to discontinue a trading relationship (done outside of the network).</p> <p>Continues with...</p> <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Performs validations.</td></tr><tr><td>2</td><td>SDP</td><td>Terminates the relationship by sending a price synchronisation message with a document command of "CHANGE_BY_REFRESH" with a relationship section action code of "CHANGE_BY_REFRESH" to the RDP and a populated relationship end effective date.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Performs validations. | 2 | SDP | Terminates the relationship by sending a price synchronisation message with a document command of "CHANGE_BY_REFRESH" with a relationship section action code of "CHANGE_BY_REFRESH" to the RDP and a populated relationship end effective date. |
| Step # | Actor | Activity Step | | | | | | | | |
| 1 | SDP | Performs validations. | | | | | | | | |
| 2 | SDP | Terminates the relationship by sending a price synchronisation message with a document command of "CHANGE_BY_REFRESH" with a relationship section action code of "CHANGE_BY_REFRESH" to the RDP and a populated relationship end effective date. | | | | | | | | |

| | | | |
|---|---|----------------|---|
| | 3 | RDP | Receives price synchronisation message and sends relationship information to data recipient. |
| | 4 | Data Recipient | Receives the trading relationship information and confirms the relationship change by responding with a an “acceptRECEIVED” response. The confirmation response message is sent to the RDP. |
| | 5 | RDP | Sends the confirmation response message to the SDP. |
| | 6 | SDP | Sends confirmation information to the data source. |
| Ends when... data source receives the confirmation response. | | | |
| Alternative Scenario | <i>The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps</i> | | |
| | Step # | Actor | Activity Step |
| | | All | The scenario shows the most anticipated choreography where the SDP sends to the RDP; but the SDP may send directly to the data recipient in situations where the SDP is also the data recipient's RDP. To reduce complexity the later is not shown in the activity steps in any scenario. |
| | 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED . See related requirements below for status codes and their actions. |
| Related Requirements | Not Applicable | | |
| Related Rules | <ol style="list-style-type: none"> Confirmation status codes other than RECEIVEDaccept are: review REVIEW – message received and no action taken yet; synchronized – message received and implemented into the backend system; REJECTEDreject – message received and terms of a specific price message segment were rejected or the data recipient wishes to terminate the price synchronisation relationship. A confirmation status of “REJECTED” is not valid for the End Trading Relationship use case. A confirmation status of “REJECTED” implies that the data recipient initiated the termination of the trading relationship. If the Document Header is “CHANGE_BY_REFRESH”, the Price Document ID must be greater than “1” The “Delete” action code implies all data associated with the Relationship ID is no longer valid only for a relationship that has not taken effect. If the relationship is already in effect, the data source must send a Modify transaction (CHANGE_BY_REFRESH) and populate the End Effective Date. In order to end a relationship segment, all dependent condition and price type segments need to be deleted/end dated before a delete/end date can be sent for the Relationship Segment. Can only end at a Relationship Segment ID level (i.e. if you have 3 relationship IDs identified for a trading relationship, in order to end the ENTIRE relationship, all 3 relationship IDs must be deleted/end dated). Reason code is conditional on the confirmation status being “Review”. If reason code is present, ensure that confirmation status in “Review”. For the Price Synchronisation Message, the Segment Action Code of “CHANGE_BY_REFRESH” assumes full refresh of the message segment only. | | |

3.6. Synchronise Conditions



3.7. Add Condition

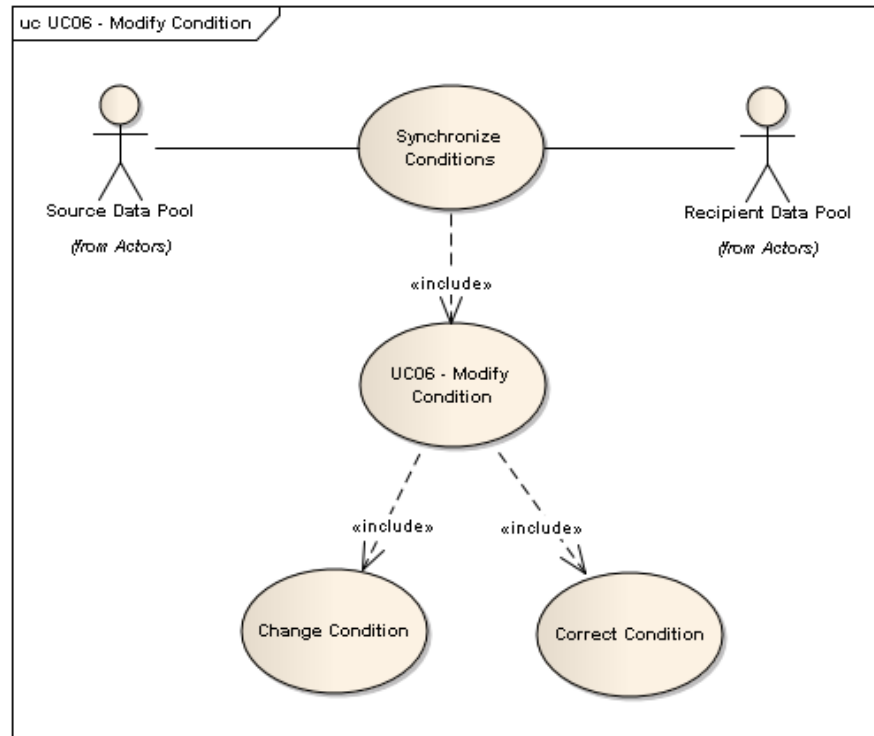


| | |
|-----------------------------|--|
| Use Case ID | UC-5 |
| Use Case Name | Add Condition |
| Use Case Description | This use case establishes non-line item conditions and summary conditions. |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient |
| Performance Goals | Establish conditions for price synchronisation. |
| Preconditions | Price synchronisation relationship has been established and price synchronisation is active. |
| Post conditions | Conditions are synchronized. |

| Scenario | Begins when... The data source notifies their SDP of price components to be added for a relationship (done outside of the network). | | | | | | | | | | | | | | | | | | | | | |
|---|---|--|--|---------------|---|----------------|--|---|-----|--|---|-----|--|---|----------------|--|---|-----|---|---|-----|--|
| | Continues with... | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Performs validations.</td></tr><tr><td>2</td><td>SDP</td><td>Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" (if the trading relationship has already been established) and the condition segment with a segment action code of "add" to the RDP, indicates the condition types and updates the price synchronisation list.</td></tr><tr><td>3</td><td>RDP</td><td>Sends the price message to the data recipient.</td></tr><tr><td>4</td><td>Data Recipient</td><td>Receives the message and confirms the conditions by responding with an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP.</td></tr><tr><td>5</td><td>RDP</td><td>Sends the confirmation response message to the SDP.</td></tr><tr><td>6</td><td>SDP</td><td>Updates the price synchronisation list and sends the confirmation response message to the data source.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Performs validations. | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" (if the trading relationship has already been established) and the condition segment with a segment action code of "add" to the RDP, indicates the condition types and updates the price synchronisation list. | 3 | RDP | Sends the price message to the data recipient. | 4 | Data Recipient | Receives the message and confirms the conditions by responding with an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP. | 5 | RDP | Sends the confirmation response message to the SDP. | 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. |
| | Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | |
| | 1 | SDP | Performs validations. | | | | | | | | | | | | | | | | | | | |
| | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" (if the trading relationship has already been established) and the condition segment with a segment action code of "add" to the RDP, indicates the condition types and updates the price synchronisation list. | | | | | | | | | | | | | | | | | | | |
| | 3 | RDP | Sends the price message to the data recipient. | | | | | | | | | | | | | | | | | | | |
| 4 | Data Recipient | Receives the message and confirms the conditions by responding with an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP. | | | | | | | | | | | | | | | | | | | | |
| 5 | RDP | Sends the confirmation response message to the SDP. | | | | | | | | | | | | | | | | | | | | |
| 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. | | | | | | | | | | | | | | | | | | | | |
| Ends when... conditions and bracket qualifiers (as needed) have been synchronized. | | | | | | | | | | | | | | | | | | | | | | |
| Alternative Scenario | <i>The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps</i> | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>3</td><td>Data Recipient</td><td>Data recipient responds with a confirmation status other than acceptRECEIVED. See related requirements below for status codes and their actions.</td></tr></table> | Step # | Actor | Activity Step | 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED. See related requirements below for status codes and their actions. | | | | | | | | | | | | | | | |
| Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | | |
| 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED. See related requirements below for status codes and their actions. | | | | | | | | | | | | | | | | | | | | |
| Related Requirements | Not Applicable | | | | | | | | | | | | | | | | | | | | | |
| Related Rules | <div><div>1.</div><div>When condition type equals bracket the bracket sub-section is used to identify the bracket qualifiers.</div></div> <div><div>2.</div><div>Header segment is mandatory and must be sent with this message.</div></div> <div><div>3.</div><div>In the condition segment, confirmations apply to the condition type and apply to all Catalogue Items or EANUCC Classification Category Codes in their respective lists.</div></div> <div><div>4.</div><div>If there has been no response to relationship segment, the condition segment is still sent to the data recipient.</div></div> <div><div>5.</div><div>Cannot send condition if relationship has been rejected.</div></div> <div><div>6.</div><div>If the Condition Segment is sent in the first price message establishing the trading relationship, the Document Header Command = ADD.</div></div> <div><div>7.</div><div>If the Condition Segment is sent after establishing the trading relationship, the Document Header Command = CHANGE_BY_REFRESH.</div></div> <div><div>8.</div><div>If Document Header Command = ADD, then Price Document ID must = "1".</div></div> <div><div>9.</div><div>If Document Header Command = CHANGE_BY_REFRESH, then Price Document ID must be greater than "1".</div></div> <div><div>10.</div><div>Confirmation status codes other than RECEIVEDaccept are: REVIEWreview – message received and no action taken yet; SYNCHRONISEDsynchronized – message received and implemented into the backend system. If no response is made the SDP will stop price synchronisation.</div></div> <div><div>11.</div><div>Confirmation Statuses (assumes one response per condition)<div><div>•</div><div>Received</div></div><div><div>•</div><div>Review</div></div></div></div> | | | | | | | | | | | | | | | | | | | | | |

| | |
|--|---|
| | <ul style="list-style-type: none"> • Synchronised • No Response- no further price synchronisation may occur <ul style="list-style-type: none"> ○ For the specified condition ID ○ Nor any price type referring to that condition ID <p>12. To depict a line item allowance in the condition the Catalogue Items(s) or Global Product Classification “brick” code(s) must be specified in the condition segment.</p> <p>13. The data recipient can override a previous confirmation status with another one through a confirmation response.</p> <p>14. Multiple confirmations can be sent by data recipients for a single price message or message segment. For example, a data recipient can send a status of ‘Received’ followed by ‘Synchronised’.</p> <p>15. A confirmation Status of Rejected is not valid for conditions.</p> <p>16. Reason code is conditional on the confirmation status being “Review”. If reason code is present, ensure that confirmation status in “Review”.</p> |
|--|---|

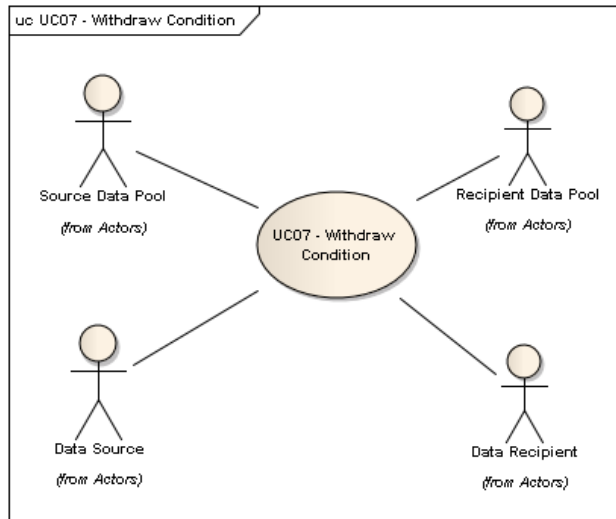
3.8. Modify Condition



| Use Case ID | UC-6 | | | | | | | | | | | | |
|----------------------|--|---|-------|---------------|---|-----|---------------------------------|---|-----|---|---|-----|--|
| Use Case Name | Modify Condition | | | | | | | | | | | | |
| Use Case Description | This use case modifies or corrects an existing non-line item conditions and summary conditions. | | | | | | | | | | | | |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient | | | | | | | | | | | | |
| Performance Goals | Change by refresh or correct conditions for price synchronisation. | | | | | | | | | | | | |
| Preconditions | Price synchronisation relationship exists and price component has been accepted by data source. | | | | | | | | | | | | |
| Post conditions | Condition has been modified. | | | | | | | | | | | | |
| Scenario | <p>Begins when... The data source notifies their SDP of modifications to item depictions and/or any related price types. (Done outside of the network).</p> <p>Continues with...</p> <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Performs necessary validations.</td></tr><tr><td>2</td><td>SDP</td><td>Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the condition segment with a segment action code of "CHANGE_BY_REFRESH" for a modification or "Correct" for a correct to the RDP, indicates the condition types and updates the price synchronisation list.</td></tr><tr><td>3</td><td>RDP</td><td>Sends the price message to the data recipient.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Performs necessary validations. | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the condition segment with a segment action code of "CHANGE_BY_REFRESH" for a modification or "Correct" for a correct to the RDP, indicates the condition types and updates the price synchronisation list. | 3 | RDP | Sends the price message to the data recipient. |
| Step # | Actor | Activity Step | | | | | | | | | | | |
| 1 | SDP | Performs necessary validations. | | | | | | | | | | | |
| 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the condition segment with a segment action code of "CHANGE_BY_REFRESH" for a modification or "Correct" for a correct to the RDP, indicates the condition types and updates the price synchronisation list. | | | | | | | | | | | |
| 3 | RDP | Sends the price message to the data recipient. | | | | | | | | | | | |

| | <table><tr><td>4</td><td>Data Recipient</td><td>Receives the message and confirms the conditions by responding with an AcceptRECEIVED confirmation response. The confirmation response message is sent to the RDP.</td></tr><tr><td>5</td><td>RDP</td><td>Sends the confirmation response message to the SDP.</td></tr><tr><td>6</td><td>SDP</td><td>Updates the price synchronisation list and sends the confirmation response message to the data source.</td></tr></table> | 4 | Data Recipient | Receives the message and confirms the conditions by responding with an Accept RECEIVED confirmation response. The confirmation response message is sent to the RDP. | 5 | RDP | Sends the confirmation response message to the SDP. | 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. |
|-----------------------------|---|---|----------------|---|---|----------------|--|---|-----|--|
| 4 | Data Recipient | Receives the message and confirms the conditions by responding with an Accept RECEIVED confirmation response. The confirmation response message is sent to the RDP. | | | | | | | | |
| 5 | RDP | Sends the confirmation response message to the SDP. | | | | | | | | |
| 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. | | | | | | | | |
| | Ends when... conditions and bracket qualifiers (as needed) have been modified. | | | | | | | | | |
| Alternative Scenario | <p><i>The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps</i></p> <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>3</td><td>Data Recipient</td><td>Data recipient responds with a confirmation status other than acceptRECEIVED. See related requirements below for status codes and their actions.</td></tr></table> | Step # | Actor | Activity Step | 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED . See related requirements below for status codes and their actions. | | | |
| Step # | Actor | Activity Step | | | | | | | | |
| 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED . See related requirements below for status codes and their actions. | | | | | | | | |
| Related Requirements | Not Applicable | | | | | | | | | |
| Related Rules | <ol style="list-style-type: none">When condition type equals bracket the bracket sub-section is used to identify the bracket qualifiers.Header segment is mandatory and must be sent with this message.In the condition segment, confirmations apply to the condition type and apply to all Catalogue Items or EANUCC Classification Category Codes in their respective lists.If there has been no response to relationship segment, the condition segment is still sent to the data recipient.Cannot send condition if relationship has been rejected.If Document Header Command = CHANGE_BY_REFRESH, then Price Document ID must be greater than “1”.Confirmation status codes other than RECEIVEDaccept are: REVIEWreview – message received and no action taken yet; SYNCHRONISEDsynchronized – message received and implemented into the backend system.If no response is made the SDP will stop price synchronisation.Condition Value Type cannot be modified.Start Effective Date cannot be modified.Start Effective Date can be corrected if it is not yet in effect (future date).If a revised Start Effective Date is required for a condition that is not yet in effect, the condition must be deleted or corrected. If a revised Start Effective Date is required for a condition that is in effect, then you must set the End Effective Date and send in a new condition with a new Start Effective Date.The data recipient can override a previous confirmation status with another one through a confirmation response.Multiple confirmations can be sent by data recipients for a single price message or message segment. For example, a data recipient can send a status of ‘Received’ followed by ‘Synchronised’.Can’t send an update for a segment if the previous add/change by refresh/correct has had no response or has been rejected.A confirmation Status of Rejected is not valid for conditions.Reason code is conditional on the confirmation status being “Review”. If reason code is present, ensure that confirmation status in “Review”.For the Price Synchronisation Message, the Segment Action Code of “CHANGE_BY_REFRESH” assumes full refresh of the message segment only. | | | | | | | | | |

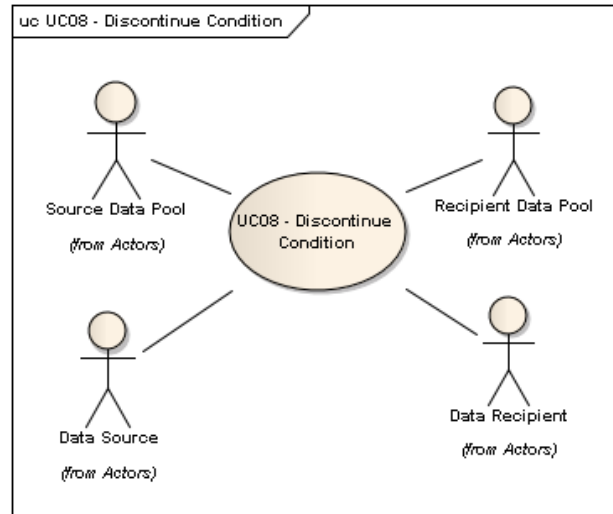
3.9. Withdraw Condition



| Use Case ID | UC-7 | | | | | | | | | | | | | | | | | | | | | |
|----------------------|--|--|--|---------------|---|-----|---------------------------------|---|-----|--|---|-----|--|---|----------------|--|---|-----|---|---|-----|--|
| Use Case Name | Withdraw Condition | | | | | | | | | | | | | | | | | | | | | |
| Use Case Description | This use case deletes an existing non-line item conditions and/or summary conditions prior to the effective start date of the condition. | | | | | | | | | | | | | | | | | | | | | |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient | | | | | | | | | | | | | | | | | | | | | |
| Performance Goals | Withdraw a condition for price synchronisation. | | | | | | | | | | | | | | | | | | | | | |
| Preconditions | Price synchronisation relationship exists and price component has been accepted by data source. | | | | | | | | | | | | | | | | | | | | | |
| Post conditions | Condition has been withdrawn. | | | | | | | | | | | | | | | | | | | | | |
| Scenario | Begins when... The data source notifies their SDP of a need to withdraw a price component (done outside of the network). | | | | | | | | | | | | | | | | | | | | | |
| | Continues with... | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Performs necessary validations.</td></tr><tr><td>2</td><td>SDP</td><td>Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the condition segment with a segment action code of "Delete" to the RDP, indicates the condition types and updates the price synchronisation list.</td></tr><tr><td>3</td><td>RDP</td><td>Sends the price message to the data recipient.</td></tr><tr><td>4</td><td>Data Recipient</td><td>Receives the message and confirms the conditions by responding with an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP.</td></tr><tr><td>5</td><td>RDP</td><td>Sends the confirmation response message to the SDP.</td></tr><tr><td>6</td><td>SDP</td><td>Updates the price synchronisation list and sends the confirmation response message to the data source.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Performs necessary validations. | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the condition segment with a segment action code of "Delete" to the RDP, indicates the condition types and updates the price synchronisation list. | 3 | RDP | Sends the price message to the data recipient. | 4 | Data Recipient | Receives the message and confirms the conditions by responding with an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP. | 5 | RDP | Sends the confirmation response message to the SDP. | 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. |
| | Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | |
| | 1 | SDP | Performs necessary validations. | | | | | | | | | | | | | | | | | | | |
| | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the condition segment with a segment action code of "Delete" to the RDP, indicates the condition types and updates the price synchronisation list. | | | | | | | | | | | | | | | | | | | |
| | 3 | RDP | Sends the price message to the data recipient. | | | | | | | | | | | | | | | | | | | |
| | 4 | Data Recipient | Receives the message and confirms the conditions by responding with an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP. | | | | | | | | | | | | | | | | | | | |
| 5 | RDP | Sends the confirmation response message to the SDP. | | | | | | | | | | | | | | | | | | | | |
| 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | |

| | | | |
|-----------------------------|---|----------------|--|
| | Ends when... conditions and bracket qualifiers (as needed) have been withdrawn. | | |
| Alternative Scenario | <i>The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps</i> | | |
| | Step # | Actor | Activity Step |
| | 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED . See related requirements below for status codes and their actions. |
| Related Requirements | Not Applicable | | |
| Related Rules | <ol style="list-style-type: none"> Header segment is mandatory and must be sent with this message. In the condition segment, confirmations apply to the condition type and apply to all Catalogue Items or EANUCC Classification Category Codes in their respective lists. If there has been no response to relationship segment, the condition segment is still sent to the data recipient. Cannot send condition if relationship has been rejected. If Document Header Command = CHANGE_BY_REFRESH, then Price Document ID must be greater than "1". A confirmation status of "REJECTED" is not valid for a withdraw. No response means that no further synchronisation can occur. The data recipient can override a previous confirmation status with another one through a confirmation response. Multiple confirmations can be sent by data recipients for a single price message or message segment. For example, a data recipient can send a status of 'Received' followed by 'Synchronised'. Reason code is conditional on the confirmation status being "Review". If reason code is present, ensure that confirmation status in "Review". For the Price Synchronisation Message, the Segment Action Code of "CHANGE_BY_REFRESH" assumes full refresh of the message segment only. | | |

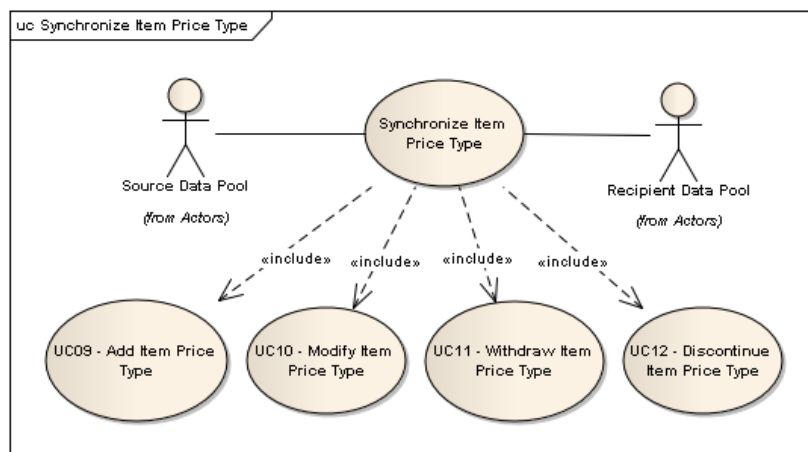
3.10. Discontinue a Condition



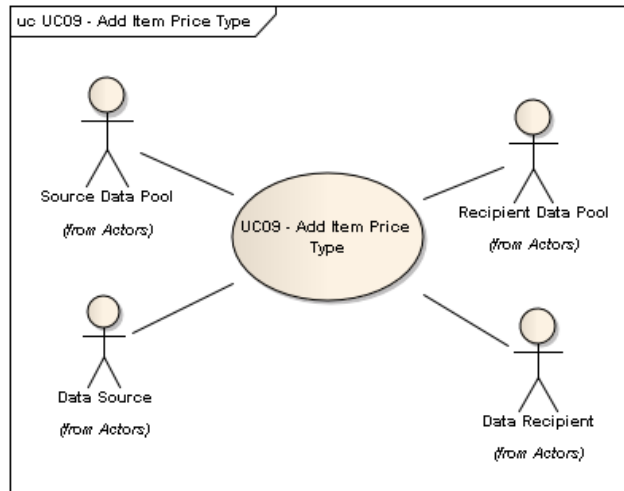
| Use Case ID | UC-8 | | | | | | | | | | | | | | | | | | |
|----------------------|---|---|--|---------------|---|-----|---------------------------------|---|-----|--|---|-----|--|---|----------------|---|---|-----|---|
| Use Case Name | Discontinue a Condition | | | | | | | | | | | | | | | | | | |
| Use Case Description | This use case discontinues an existing non-line item conditions and/or summary conditions that are already in effect. | | | | | | | | | | | | | | | | | | |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient | | | | | | | | | | | | | | | | | | |
| Performance Goals | Discontinues a condition for price synchronisation. | | | | | | | | | | | | | | | | | | |
| Preconditions | Price synchronisation relationship exists and price component has been accepted by data source. | | | | | | | | | | | | | | | | | | |
| Post conditions | Condition has been discontinue. | | | | | | | | | | | | | | | | | | |
| Scenario | Begins when... The data source notifies their SDP of a need to discontinue a price component. (Done outside of the network). | | | | | | | | | | | | | | | | | | |
| | Continues with... | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Performs necessary validations.</td></tr><tr><td>2</td><td>SDP</td><td>Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH", the condition segment with a segment action code of "CHANGE_BY_REFRESH" and an updated Condition End Effective Date to the RDP, indicates the condition types and updates the price synchronisation list.</td></tr><tr><td>3</td><td>RDP</td><td>Sends the price message to the data recipient.</td></tr><tr><td>4</td><td>Data Recipient</td><td>Receives the message and confirms the conditions by responding with an accept RECEIVED confirmation response. The confirmation response message is sent to the RDP.</td></tr><tr><td>5</td><td>RDP</td><td>Sends the confirmation response message to the SDP.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Performs necessary validations. | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH", the condition segment with a segment action code of "CHANGE_BY_REFRESH" and an updated Condition End Effective Date to the RDP, indicates the condition types and updates the price synchronisation list. | 3 | RDP | Sends the price message to the data recipient. | 4 | Data Recipient | Receives the message and confirms the conditions by responding with an accept RECEIVED confirmation response. The confirmation response message is sent to the RDP. | 5 | RDP | Sends the confirmation response message to the SDP. |
| | Step # | Actor | Activity Step | | | | | | | | | | | | | | | | |
| | 1 | SDP | Performs necessary validations. | | | | | | | | | | | | | | | | |
| | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH", the condition segment with a segment action code of "CHANGE_BY_REFRESH" and an updated Condition End Effective Date to the RDP, indicates the condition types and updates the price synchronisation list. | | | | | | | | | | | | | | | | |
| | 3 | RDP | Sends the price message to the data recipient. | | | | | | | | | | | | | | | | |
| 4 | Data Recipient | Receives the message and confirms the conditions by responding with an accept RECEIVED confirmation response. The confirmation response message is sent to the RDP. | | | | | | | | | | | | | | | | | |
| 5 | RDP | Sends the confirmation response message to the SDP. | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | |

| | <table><tr><td>6</td><td>SDP</td><td>Updates the price synchronisation list and sends the confirmation response message to the data source.</td></tr></table> | 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. | | | |
|--|--|--|-------|--|---|----------------|--|
| 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. | | | | | |
| Ends when... conditions and bracket qualifiers (as needed) have been discontinued. | | | | | | | |
| Alternative Scenario | The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>3</td><td>Data Recipient</td><td>Data recipient responds with a confirmation status other than <u>RECEIVED</u>accept. See related requirements below for status codes and their actions.</td></tr></table> | Step # | Actor | Activity Step | 3 | Data Recipient | Data recipient responds with a confirmation status other than <u>RECEIVED</u> accept . See related requirements below for status codes and their actions. |
| Step # | Actor | Activity Step | | | | | |
| 3 | Data Recipient | Data recipient responds with a confirmation status other than <u>RECEIVED</u> accept . See related requirements below for status codes and their actions. | | | | | |
| Related Rules | <ol style="list-style-type: none">Header segment is mandatory and must be sent with this message.In the condition segment, confirmations apply to the condition type and apply to all Catalogue Items or EANUCC Classification Category Codes in their respective lists.If there has been no response to relationship segment, the condition segment is still sent to the data recipient.Cannot send condition if relationship has been rejected.If Document Header Command = CHANGE_BY_REFRESH, then Price Document ID must be greater than “1”.No response means that no further synchronisation can occur.The data recipient can override a previous confirmation status with another one through a confirmation response.Multiple confirmations can be sent by data recipients for a single price message or message segment. For example, a data recipient can send a status of ‘Received’ followed by ‘Synchronised’.Reason code is conditional on the confirmation status being “Review”. If reason code is present, ensure that confirmation status in “Review”.For the Price Synchronisation Message, the Segment Action Code of “CHANGE_BY_REFRESH” assumes full refresh of the message segment only. | | | | | | |

3.11. Synchronise Price Type



3.12. Add Item Price Type



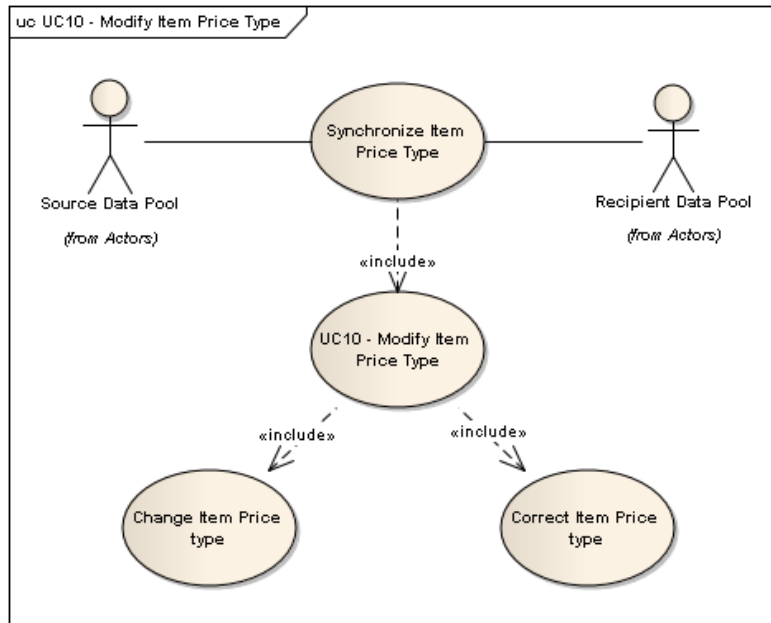
| Use Case ID | UC-9 | | | | | | | | | | | | | | | | | | | | | |
|----------------------|--|--|-------|---------------|---|-----|---------------------------------|---|-----|--|---|-----|--|---|----------------|--|---|-----|---|---|-----|--|
| Use Case Name | Add Item Price Type | | | | | | | | | | | | | | | | | | | | | |
| Use Case Description | This use case establishes an item depiction and any related price types. | | | | | | | | | | | | | | | | | | | | | |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient | | | | | | | | | | | | | | | | | | | | | |
| Performance Goals | Establish an item depiction and all related price types for price synchronisation. | | | | | | | | | | | | | | | | | | | | | |
| Preconditions | Price synchronisation relationship has been established and price synchronisation is active. | | | | | | | | | | | | | | | | | | | | | |
| Post conditions | Item Depictions and related price types are synchronized. | | | | | | | | | | | | | | | | | | | | | |
| Scenario | <p>Begins when... The data source notifies their SDP of new price types that they want to be synchronised with a trading partner. (Done outside of the network).</p> <p>Continues with...</p> <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Performs necessary validations.</td></tr><tr><td>2</td><td>SDP</td><td>Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" (if the trading relationship has already been established) and the item depiction and price type segments with a segment action code of "add" to the RDP and updates the price synchronisation list.</td></tr><tr><td>3</td><td>RDP</td><td>Sends the price message to the data recipient.</td></tr><tr><td>4</td><td>Data Recipient</td><td>Receives the message and confirms the item depiction and price type segments by responding with an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP.</td></tr><tr><td>5</td><td>RDP</td><td>Sends the confirmation response message to the SDP.</td></tr><tr><td>6</td><td>SDP</td><td>Updates the price synchronisation list and sends the confirmation response message to the data source.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Performs necessary validations. | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" (if the trading relationship has already been established) and the item depiction and price type segments with a segment action code of "add" to the RDP and updates the price synchronisation list. | 3 | RDP | Sends the price message to the data recipient. | 4 | Data Recipient | Receives the message and confirms the item depiction and price type segments by responding with an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP. | 5 | RDP | Sends the confirmation response message to the SDP. | 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. |
| Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | | |
| 1 | SDP | Performs necessary validations. | | | | | | | | | | | | | | | | | | | | |
| 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" (if the trading relationship has already been established) and the item depiction and price type segments with a segment action code of "add" to the RDP and updates the price synchronisation list. | | | | | | | | | | | | | | | | | | | | |
| 3 | RDP | Sends the price message to the data recipient. | | | | | | | | | | | | | | | | | | | | |
| 4 | Data Recipient | Receives the message and confirms the item depiction and price type segments by responding with an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP. | | | | | | | | | | | | | | | | | | | | |
| 5 | RDP | Sends the confirmation response message to the SDP. | | | | | | | | | | | | | | | | | | | | |
| 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. | | | | | | | | | | | | | | | | | | | | |

| | Ends when... item Depictions and related price types have been synchronized. | | | | | | |
|-----------------------------|--|--|-------|---------------|---|----------------|--|
| Alternative Scenario | <p><i>The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps</i></p> <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>3</td><td>Data Recipient</td><td>Data recipient responds with a confirmation status other than acceptRECEIVED. See related requirements below for status codes and their actions.</td></tr></table> | Step # | Actor | Activity Step | 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED . See related requirements below for status codes and their actions. |
| Step # | Actor | Activity Step | | | | | |
| 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED . See related requirements below for status codes and their actions. | | | | | |
| Related Requirements | Not Applicable | | | | | | |
| Related Rules | <ol style="list-style-type: none">1. The Item Price Segment is mandatory if the Item Depiction qualifier has been populated.2. Header segment is mandatory and must be sent with this message.3. Relationship Segment and Condition Segment is not needed in a price synchronisation message to send Item Depiction and Price Type.4. Cannot send Item Depiction and Price Type if the Trading Relationship has been rejected.5. A targeted condition in the Item Price Type must have a prior confirmation status of RECEIVEDaccept, synchronize or review except when a segment is first synchronized (as an Add). In this case, it may also be targeted in the same file without the requirement of a prior confirmation status.6. A targeted price type in the Item Price Type segment must have a prior confirmation status of RECEIVEDaccept, SYNCHRONISEDsynchronize or reviewREVIEW except when a segment is first synchronized (as an Add). In this case, it may also be targeted in the same file without the requirement of a prior confirmation status.7. Multiple Price Types may exist simultaneously for a Catalogue Item and each can have their own confirmation status8. Bracket Qualifiers for a Price Type can be sent providing that the brackets have not been sent as standard brackets in the condition segment.9. If Document Header Command = ADD, then Price Document ID must = "1".10. Confirmation status codes other than RECEIVEDaccept are: REVIEWreview – message received and no action taken yet; SYNCHRONISEDsynchronized – message received and implemented into the backend system; reject – message received and the data recipient does not wish to receive the price information for the given GTIN. If no response is made the SDP will stop price synchronisation.11. No Response stops further synchronisation of only the specific Price types for which the confirmation status is "No response"12. A confirmation status of "Rejected" stops further synchronisation of all Price Types for the associated trade item. This rule is not valid for the Restart Process which has different functionality around the Rejected Status (see Implementation Considerations).13. If the Ship To or Ship From Business Location attributes are populated with more than one value (a list), the data recipient may only accept or reject all, not individually by Business Location14. Condition segment is not required.15. Relationship segment is not required.16. At least one Item Price Segment is mandatory if the Item Depiction qualifier has been populated.17. Can only populate Target Price type if "Price Type" is equal to "Allowance" or "Charge" or "Promotional Price" or any of the "Transactional Price Types."18. If the Target Price Type is not populated, the allowance/charge is tied to the catalogue item itself; regardless of the base price it is using (i.e. will span all base price brackets).19. Data recipients cannot accept or reject individual brackets qualifiers.20. When an Item Price Type refers to a condition: The Condition Type must be of type "Bracket" | | | | | | |

The Item Price segment must only be of Price type "Bracket".

21. The data recipient can override a previous confirmation status with another one through a confirmation response. Exception: cannot do this with a Price Type that has been rejected.
22. Multiple confirmations can be sent by data recipients for a single price message or message segment. For example, a data recipient can send a status of 'Received' followed by 'Synchronised'. Exception: a data recipient cannot modify a status of Rejected. A Restart is the only way to re-initiate synchronisation on a Price Type that has been rejected.
23. Reason code is conditional on the confirmation status being "Review". If reason code is present, ensure that confirmation status in "Review".
24. For the Price Synchronisation Message, the Segment Action Code of "CHANGE_BY_REFRESH" assumes full refresh of the message segment only.
Any rejection of a targeted price type would result in the rejection of the targeting price type.

3.13. Modify Item Price Type

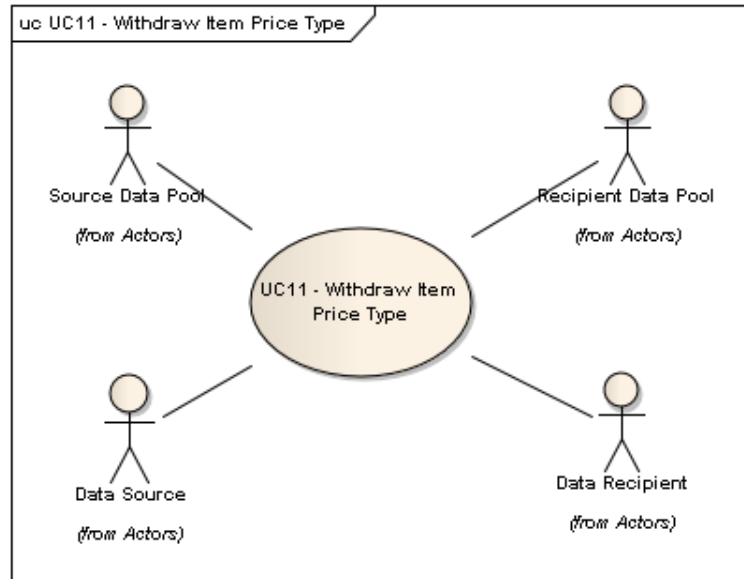


| | |
|-----------------------------|---|
| Use Case ID | UC-10 |
| Use Case Name | Modify Item Price Type |
| Use Case Description | This use case modifies or corrects existing item depictions and/or any related price types. |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient |
| Performance Goals | Change by refresh or correct item depictions and/or any related price types used for price synchronisation. |
| Preconditions | Price synchronisation relationship exists and item depictions and/or any related price types have been accepted by data source. |
| Post conditions | Item depictions and/or any related price types have been modified. |

| Scenario | Begins when... The data source notifies their SDP of modifications to price components (done outside of the network). | | | | | | | | | | | | | | | | | | | | | |
|--|---|--|---|---------------|---|----------------|--|---|-----|---|---|-----|--|---|----------------|---|---|-----|---|---|-----|--|
| | Continues with... | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Performs necessary validations.</td></tr><tr><td>2</td><td>SDP</td><td>Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the item depiction and price type segments with a segment action code of "CHANGE_BY_REFRESH" for modification and "Correct" for a correction to the RDP and updates the price synchronisation list.</td></tr><tr><td>3</td><td>RDP</td><td>Sends the price message to the data recipient.</td></tr><tr><td>4</td><td>Data Recipient</td><td>Receives the message and confirms the item depiction and price type segments by responding with a an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP.</td></tr><tr><td>5</td><td>RDP</td><td>Sends the confirmation response message to the SDP.</td></tr><tr><td>6</td><td>SDP</td><td>Updates the price synchronisation list and sends the confirmation response message to the data source.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Performs necessary validations. | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the item depiction and price type segments with a segment action code of "CHANGE_BY_REFRESH" for modification and "Correct" for a correction to the RDP and updates the price synchronisation list. | 3 | RDP | Sends the price message to the data recipient. | 4 | Data Recipient | Receives the message and confirms the item depiction and price type segments by responding with a an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP. | 5 | RDP | Sends the confirmation response message to the SDP. | 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. |
| | Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | |
| | 1 | SDP | Performs necessary validations. | | | | | | | | | | | | | | | | | | | |
| | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the item depiction and price type segments with a segment action code of "CHANGE_BY_REFRESH" for modification and "Correct" for a correction to the RDP and updates the price synchronisation list. | | | | | | | | | | | | | | | | | | | |
| | 3 | RDP | Sends the price message to the data recipient. | | | | | | | | | | | | | | | | | | | |
| | 4 | Data Recipient | Receives the message and confirms the item depiction and price type segments by responding with a an "acceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP. | | | | | | | | | | | | | | | | | | | |
| 5 | RDP | Sends the confirmation response message to the SDP. | | | | | | | | | | | | | | | | | | | | |
| 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. | | | | | | | | | | | | | | | | | | | | |
| Ends when... the item depiction and price type segments have been modified. | | | | | | | | | | | | | | | | | | | | | | |
| Alternative Scenario | <i>The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps</i> | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>3</td><td>Data Recipient</td><td>Data recipient responds with a confirmation status other than acceptRECEIVED. See related requirements below for status codes and their actions.</td></tr></table> | Step # | Actor | Activity Step | 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED. See related requirements below for status codes and their actions. | | | | | | | | | | | | | | | |
| Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | | |
| 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED. See related requirements below for status codes and their actions. | | | | | | | | | | | | | | | | | | | | |
| Related Requirements | Not Applicable | | | | | | | | | | | | | | | | | | | | | |
| Related Rules | <div><div>1.</div><div>Header segment is mandatory and must be sent with this message.</div></div> <div><div>2.</div><div>If Document Header Command = CHANGE_BY_REFRESH, then Price Document ID must be greater than "1".</div></div> <div><div>3.</div><div>Confirmation status codes other than RECEIVEDaccept are: REVIEWreview – message received and no action taken yet; SYNCHRONISEDsynchronize – message received and implemented into the backend system; reject – message received and the data recipient does not wish to receive the price information for the given GTIN. If no response is made the SDP will stop price synchronisation. Can only send a reject if related to a specific catalogue item. This is used in the event that the data recipient does not wish to synchronise pricing for the indicated catalogue item.</div></div> <div><div>4.</div><div>Start Effective Date cannot be modified.</div></div> <div><div>5.</div><div>Start Effective Date can be corrected if it is not yet in effect (future date).</div></div> <div><div>6.</div><div>Cannot send Item Depiction and Price Type if the Trading Relationship has been rejected.</div></div> <div><div>7.</div><div>A targeted condition in the Item Price Type must have a prior confirmation status of RECEIVEDaccept, SYNCHRONISEDsynchronize or REVIEWreview except when a segment is first synchronized (as an Add). In this case, it may also be targeted in the same file without the requirement of a prior confirmation status.</div></div> <div><div>8.</div><div>A targeted price type in the Item Price Type segment must have a prior confirmation status of RECEIVEDaccept, SYNCHRONISEDsynchronize or reviewREVIEW except when a segment is first synchronized (as an Add). In this case, it may also be targeted in the same file without the requirement of a prior confirmation status.</div></div> | | | | | | | | | | | | | | | | | | | | | |

9. Multiple Item price types may exist simultaneously for a Catalogue Item and each can have their own confirmation status
10. Bracket Qualifiers for a Price Type can be sent providing that the brackets have not been sent as standard brackets in the condition segment.
11. No Response stops further synchronisation of only the specific price types for which the confirmation status is "No response".
12. A confirmation status of "Rejected" stops further synchronisation of all Price Types for the associated trade item. This rule is not valid for the Restart Process which has different functionality around the Rejected Status (see Implementation Considerations).
13. If the Ship To or Ship From Business Location attributes are populated with more than one value (a list), the data recipient may only accept or reject all, not individually by Business Location
14. Condition segment is not required.
15. Relationship segment is not required.
16. The Item Price Segment is mandatory if the Item Depiction qualifier has been populated.
17. Can only populate Target Price type if "Price Type" is equal to "Allowance" or "Charge" or "Promotional Price" or any of the "Transactional Price Types."
18. If the target item price type is not populated, the allowance/charge is tied to the catalogue item itself; regardless of the base price it is using (i.e. will span all base price brackets).
19. Data recipients cannot accept or reject individual brackets qualifiers.
20. When an Item Price Type refers to a condition:
The Condition Type must be of type "Bracket"
The Item Price segment must only be of Price type "Bracket".
21. The data recipient can override a previous confirmation status with another one through a confirmation response. Exception: cannot do this with a Price Type that has been rejected.
22. Multiple confirmations can be sent by data recipients for a single price message or message segment. For example, a data recipient can send a status of 'RECEIVED' followed by 'SYNCHRONISED'. Exception: a data recipient cannot modify a status of REJECTED. A Restart is the only way to re-initiate synchronisation on a Price Type that has been rejected.
23. Can't send an update for a segment if the previous add/change by refresh/correct has had no response or has been rejected.
24. If a revised Start Effective Date is required for a price type that is not yet in effect, the price type must be deleted or corrected. If a revised Start Effective Date is required for a price type that is in effect, then you must set the End Effective Date and send in a new price type with a new Start Effective Date.
25. Reason code is conditional on the confirmation status being "REVIEW". If reason code is present, ensure that confirmation status in "REVIEW".
26. For the Price Synchronisation Message, the Segment Action Code of "CHANGE_BY_REFRESH" assumes full refresh of the message segment only.
27. Any rejection of a targeted price type would result in the rejection of the targeting price type.

3.14. Withdraw Item Price Type

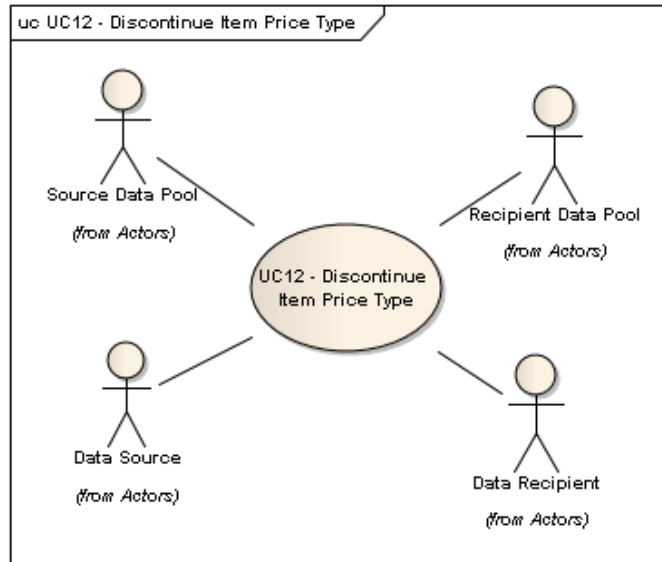


| | |
|-----------------------------|--|
| Use Case ID | UC-11 |
| Use Case Name | Withdraw Item Price Type |
| Use Case Description | This use case deletes existing item depictions and/or any related price types prior to the effective start date of the price type. |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient |
| Performance Goals | Withdraw existing item depictions and/or any related price types that are not in effect. |
| Preconditions | Price synchronisation relationship exists and item depictions and/or any related price types have been accepted by data source. |
| Post conditions | Item depictions and/or any related price types have been withdrawn. |

| Scenario | Begins when... The data source notifies their SDP of a need to withdraw item depictions and/or any related price types. (Done outside of the network). | | | | | | | | | | | | | | | | | | | | | |
|---|---|--|--|---------------|---|----------------|--|---|-----|--|---|-----|--|---|----------------|--|---|-----|---|---|-----|--|
| | Continues with... | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Performs necessary validations.</td></tr><tr><td>2</td><td>SDP</td><td>Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the item depiction and price type segments with a segment action code "Delete" and updates the price synchronisation list.</td></tr><tr><td>3</td><td>RDP</td><td>Sends the price message to the data recipient.</td></tr><tr><td>4</td><td>Data Recipient</td><td>Receives the message and confirms the withdrawal of the item depiction and price type segments by responding with an "AcceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP.</td></tr><tr><td>5</td><td>RDP</td><td>Sends the confirmation response message to the SDP.</td></tr><tr><td>6</td><td>SDP</td><td>Updates the price synchronisation list and sends the confirmation response message to the data source.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Performs necessary validations. | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the item depiction and price type segments with a segment action code "Delete" and updates the price synchronisation list. | 3 | RDP | Sends the price message to the data recipient. | 4 | Data Recipient | Receives the message and confirms the withdrawal of the item depiction and price type segments by responding with an "AcceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP. | 5 | RDP | Sends the confirmation response message to the SDP. | 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. |
| | Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | |
| | 1 | SDP | Performs necessary validations. | | | | | | | | | | | | | | | | | | | |
| | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the item depiction and price type segments with a segment action code "Delete" and updates the price synchronisation list. | | | | | | | | | | | | | | | | | | | |
| | 3 | RDP | Sends the price message to the data recipient. | | | | | | | | | | | | | | | | | | | |
| | 4 | Data Recipient | Receives the message and confirms the withdrawal of the item depiction and price type segments by responding with an "AcceptRECEIVED" confirmation response. The confirmation response message is sent to the RDP. | | | | | | | | | | | | | | | | | | | |
| 5 | RDP | Sends the confirmation response message to the SDP. | | | | | | | | | | | | | | | | | | | | |
| 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. | | | | | | | | | | | | | | | | | | | | |
| Ends when... item depictions and/or any related price types have been withdrawn. | | | | | | | | | | | | | | | | | | | | | | |
| Alternative Scenario | <i>The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps</i> | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>3</td><td>Data Recipient</td><td>Data recipient responds with a confirmation status other than accept. See related requirements below for status codes and their actions.</td></tr></table> | Step # | Actor | Activity Step | 3 | Data Recipient | Data recipient responds with a confirmation status other than accept. See related requirements below for status codes and their actions. | | | | | | | | | | | | | | | |
| Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | | |
| 3 | Data Recipient | Data recipient responds with a confirmation status other than accept. See related requirements below for status codes and their actions. | | | | | | | | | | | | | | | | | | | | |
| Related Requirements | Not Applicable | | | | | | | | | | | | | | | | | | | | | |
| Related Rules | <div><div>1.</div><div>Header segment is mandatory and must be sent with this message.</div></div> <div><div>2.</div><div>If Document Header Command = CHANGE_BY_REFRESH, then Price Document ID must be greater than "1".</div></div> <div><div>3.</div><div>A confirmation status of "REJECTED" is not valid for a withdrawal of a Price Type.</div></div> <div><div>4.</div><div>A confirmation status of "REVIEW" is not valid for a withdrawal or a Price Type.</div></div> <div><div>5.</div><div>No response means that no further synchronisation can occur.</div></div> <div><div>6.</div><div>Condition segment is not required.</div></div> <div><div>7.</div><div>Relationship segment is not required.</div></div> <div><div>8.</div><div>The Item Price Segment is mandatory if the Item Depiction qualifier has been populated.</div></div> <div><div>9.</div><div>When the Item Price Type Segment Action Code = Delete (<u>Parent Price Type</u>)<div><div>•</div><div>Must delete the parent Price Type and all children (targeted) Price Types for the Price Type being withdrawn.</div></div><div><div>•</div><div><u>—</u> This deletes all price types associated with an item but not the item.</div></div><div><div>•</div><div>Note: within the XML message, all child Price Types must be deleted prior to the parent Price Type.</div></div></div></div> <div><div>10.</div><div>When the Item Price Type Segment Action Code = Delete (Child Price Type)<div><div>•</div><div>Must delete the individual child Price Type only and not the parent Price Type.</div><div><div>•</div><div>This deletes the individual child price type associated with an item but not the item.</div></div></div></div><div><div>11.</div><div>The data recipient can override a previous confirmation status with another one through a confirmation response. Exception: cannot do this with a Price Type that has been rejected.</div></div></div> | | | | | | | | | | | | | | | | | | | | | |

- | | |
|--|--|
| | <p>12. Multiple confirmations can be sent by data recipients for a single price message or message segment. For example, a data recipient can send a status of '<u>RECEIVED</u>' followed by '<u>SYNCHRONISED</u>'. Exception: a data recipient cannot modify a status of <u>REJECTED</u>. A Restart is the only way to re-initiate synchronisation on a Price Type that has been rejected.</p> <p>13. For the Price Synchronisation Message, the Segment Action Code of "CHANGE_BY_REFRESH" assumes full refresh of the message segment only.</p> |
|--|--|

3.15. Discontinue Item Price Type



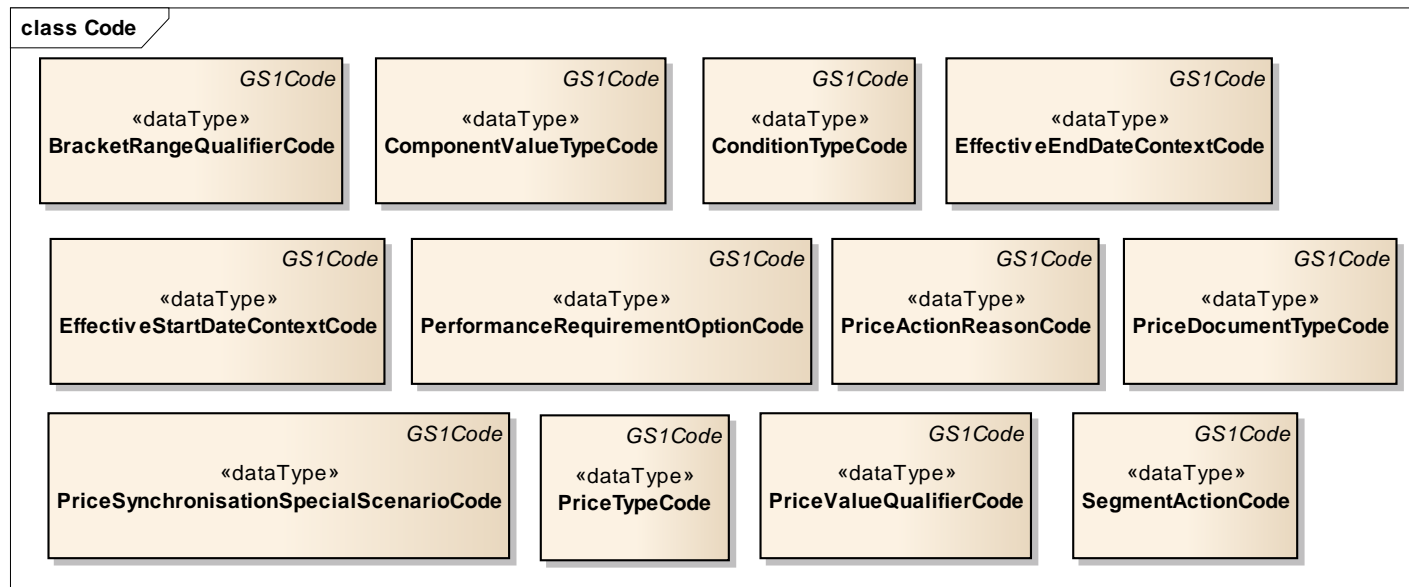
| | |
|-----------------------------|--|
| Use Case ID | UC-12 |
| Use Case Name | Discontinue Item Price Type |
| Use Case Description | This use case deletes existing item depictions and/or any related price types that are already in effect. |
| Actors (Goal) | Data source, Source Data Pool, Recipient Data Pool, Data Recipient |
| Performance Goals | Discontinues existing item depictions and/or any related price types. |
| Preconditions | Price synchronisation relationship exists and existing item depictions and/or any related price types have been accepted by data source. |
| Post conditions | Item depictions and/or any related price types have been discontinued. |

| Scenario | Begins when... The data source notifies their SDP of a need to discontinue item depictions and/or any related price types. (Done outside of the network). | | | | | | | | | | | | | | | | | | | | | |
|--|--|--|---|---------------|---|----------------|--|---|-----|---|---|-----|--|---|----------------|---|---|-----|---|---|-----|--|
| | Continues with... | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>1</td><td>SDP</td><td>Performs necessary validations.</td></tr><tr><td>2</td><td>SDP</td><td>Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the item depiction and price type segments with a segment action code "CHANGE_BY_REFRESH" and updates the price synchronisation list.</td></tr><tr><td>3</td><td>RDP</td><td>Sends the price message to the data recipient.</td></tr><tr><td>4</td><td>Data Recipient</td><td>Receives the message and confirms the discontinuation of the item depiction and price type segments by responding with an "accept" confirmation response. The confirmation response message is sent to the RDP.</td></tr><tr><td>5</td><td>RDP</td><td>Sends the confirmation response message to the SDP.</td></tr><tr><td>6</td><td>SDP</td><td>Updates the price synchronisation list and sends the confirmation response message to the data source.</td></tr></table> | Step # | Actor | Activity Step | 1 | SDP | Performs necessary validations. | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the item depiction and price type segments with a segment action code "CHANGE_BY_REFRESH" and updates the price synchronisation list. | 3 | RDP | Sends the price message to the data recipient. | 4 | Data Recipient | Receives the message and confirms the discontinuation of the item depiction and price type segments by responding with an "accept" confirmation response. The confirmation response message is sent to the RDP. | 5 | RDP | Sends the confirmation response message to the SDP. | 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. |
| | Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | |
| | 1 | SDP | Performs necessary validations. | | | | | | | | | | | | | | | | | | | |
| | 2 | SDP | Creates a price synchronisation message using document command of "CHANGE_BY_REFRESH" and the item depiction and price type segments with a segment action code "CHANGE_BY_REFRESH" and updates the price synchronisation list. | | | | | | | | | | | | | | | | | | | |
| | 3 | RDP | Sends the price message to the data recipient. | | | | | | | | | | | | | | | | | | | |
| | 4 | Data Recipient | Receives the message and confirms the discontinuation of the item depiction and price type segments by responding with an "accept" confirmation response. The confirmation response message is sent to the RDP. | | | | | | | | | | | | | | | | | | | |
| 5 | RDP | Sends the confirmation response message to the SDP. | | | | | | | | | | | | | | | | | | | | |
| 6 | SDP | Updates the price synchronisation list and sends the confirmation response message to the data source. | | | | | | | | | | | | | | | | | | | | |
| Ends when... the item depiction and price type segments have been discontinued. | | | | | | | | | | | | | | | | | | | | | | |
| Alternative Scenario | <i>The step #s below are related to the step #s in the scenario and are alternatives to the scenario steps</i> | | | | | | | | | | | | | | | | | | | | | |
| | <table><tr><th>Step #</th><th>Actor</th><th>Activity Step</th></tr><tr><td>3</td><td>Data Recipient</td><td>Data recipient responds with a confirmation status other than acceptRECEIVED. See related requirements below for status codes and their actions.</td></tr></table> | Step # | Actor | Activity Step | 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED . See related requirements below for status codes and their actions. | | | | | | | | | | | | | | | |
| | Step # | Actor | Activity Step | | | | | | | | | | | | | | | | | | | |
| 3 | Data Recipient | Data recipient responds with a confirmation status other than accept RECEIVED . See related requirements below for status codes and their actions. | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | |
| Related Rules | 1. Header segment is mandatory and must be sent with this message. | | | | | | | | | | | | | | | | | | | | | |
| | 2. If Document Header Command = CHANGE_BY_REFRESH, then Price Document ID must be greater than "1". | | | | | | | | | | | | | | | | | | | | | |
| | 3. A confirmation status of "REJECTED" is not valid for a discontinue. | | | | | | | | | | | | | | | | | | | | | |
| | 4. A confirmation status of "REVIEW" is not valid for a discontinue. | | | | | | | | | | | | | | | | | | | | | |
| | 5. No response means that no further synchronisation can occur. | | | | | | | | | | | | | | | | | | | | | |
| | 6. Condition segment is not required. | | | | | | | | | | | | | | | | | | | | | |
| | 7. Relationship segment is not required. | | | | | | | | | | | | | | | | | | | | | |
| | 8. For a discontinue, the End Effective Date must be populated or updated. | | | | | | | | | | | | | | | | | | | | | |
| | 9. The data recipient can override a previous confirmation status with another one through a confirmation response. Exception: cannot do this with a Price Type that has been rejected. | | | | | | | | | | | | | | | | | | | | | |
| | 10. Multiple confirmations can be sent by data recipients for a single price message or message segment. For example, a data recipient can send a status of 'RECEIVED' followed by 'SYNCHRONISED'. Exception: a data recipient cannot modify a status of REJECTED. A Restart is the only way to re-initiate synchronisation on a Price Type that has been rejected. | | | | | | | | | | | | | | | | | | | | | |
| | 11. For the Price Synchronisation Message, the Segment Action Code of "CHANGE_BY_REFRESH" assumes full refresh of the message segment only. | | | | | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | | | |

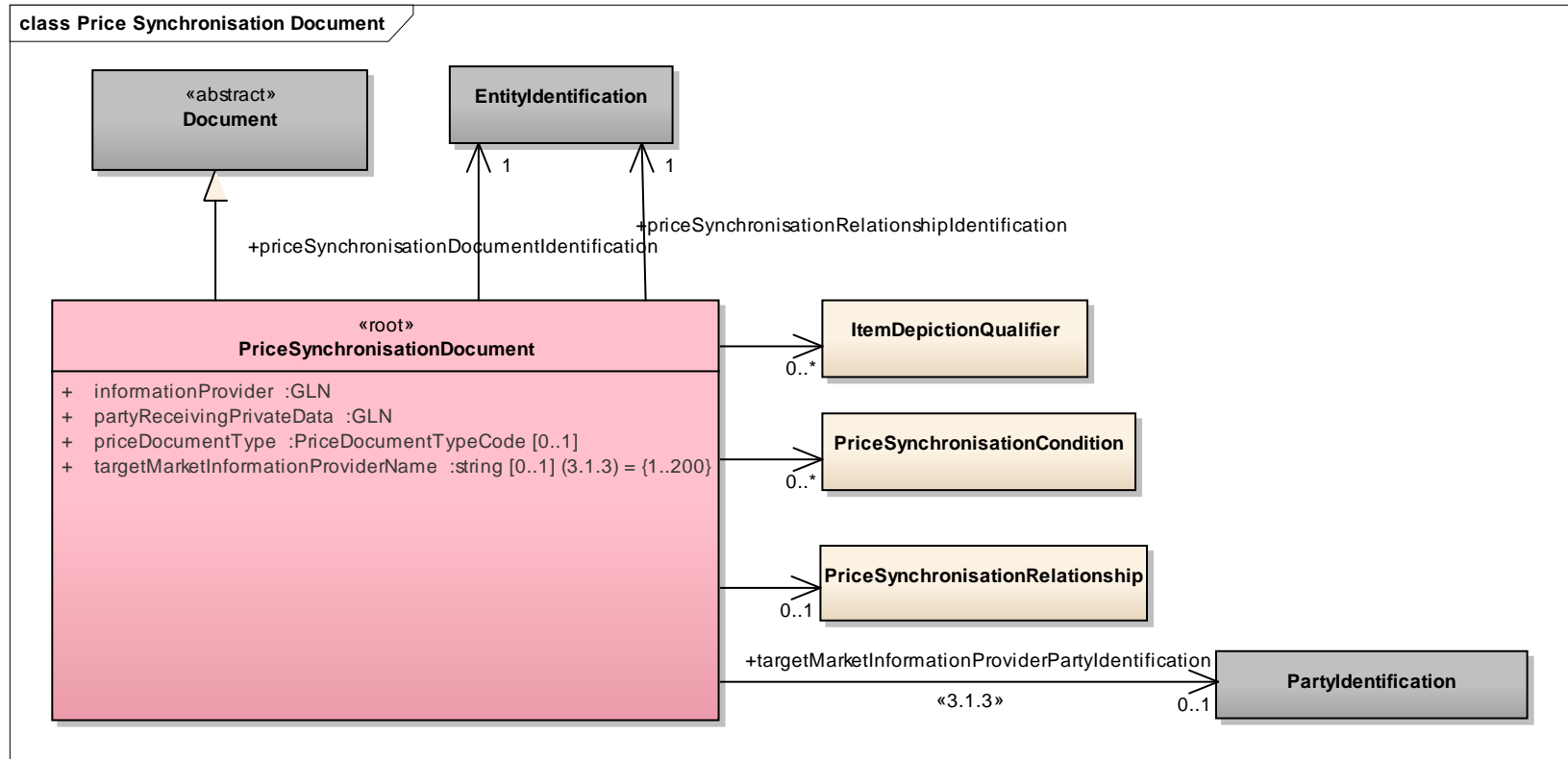
4. Information Model

4.1. Class Diagrams

4.1.1. Codes



4.1.2. Price Synchronisation Document

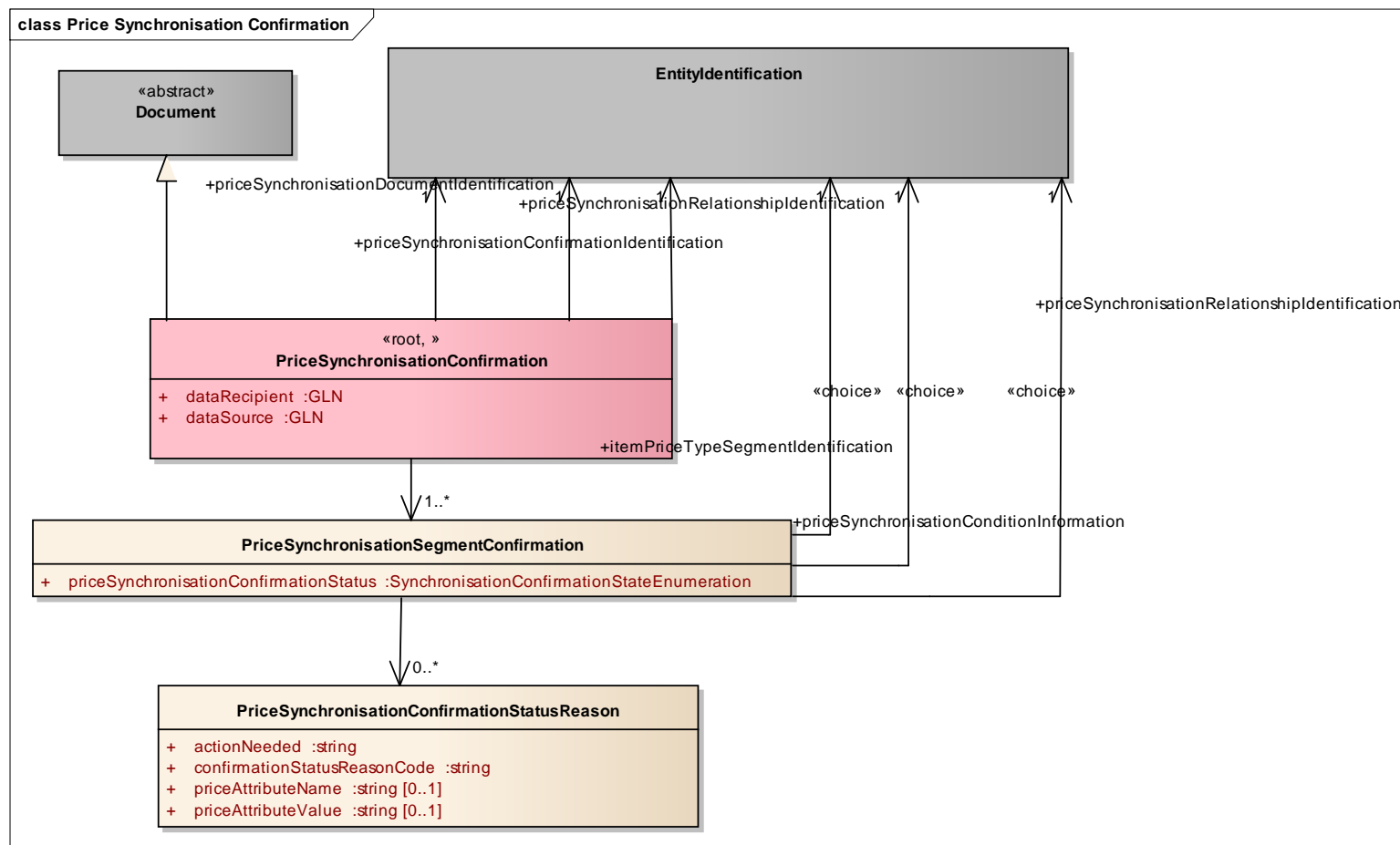


Note: Reference Shared Common Library Business Message (BMS) Release 3.1.0

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|----------------------------------|--|---------------------------|------------------|--|
| PriceSynchronisationDoc ument | | | | An electronic document used to synchronise pricing information including pricing relationship, pricing elements and item price depiction between trading partners in order to facilitate an invoice amount equal to the expected payment amount equal to the actual payment. |
| Association | priceSynchronisationRelations hipIdentification | EntityIdentification | 1..1 | A string of characters assigned by the Information Provider to uniquely identify each price synchronization relationship that exists between |

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|--------------------|--|-----------------------------------|------------------|---|
| | | | | the Information Provider and the Party Receiving Private Data. Each Price Synchronisation Message can only contain price information related to a single price synchronization relationshi |
| Association | | PriceSynchronisation RelationShip | 0..1 | Provides the depiction of a price synchronisation relationship for a price synchronisation document |
| Association | | ItemDepictionQualifie r | 0..* | Provides one or more item depictions for a price synchronisation document |
| Association | | PriceSynchronisation Condition | 0..* | Provides one or many price synchronisation conditions for a price synchronisation document |
| Association | priceSynchronisationDocumen tIdentification | EntityIdentification | 1..1 | Within a given price synchronization relationship, a number assigned by the Source Data Pool to uniquely identify each instance of a Price Synchronization Message sent from the Source Data Pool to the Party Receiving Private Data. The number is unique within each Price synchronization relationship |
| <u>Association</u> | <u>targetMarketInformationProvid erPartyIdentification</u> | <u>PartyIdentification</u> | <u>0..1</u> | <u>The party identification (GLN and additional) of any local provider of price information for an item if this party is different than the Price Synchronisation Document Information Provider.</u> |
| Generalization | | Document | | |
| Attribute | informationProvider | GLN | 1..1 | The party who owns the data. |
| Attribute | partyReceivingPrivateData | GLN | 1..1 | Party, which is authorized to view, use, download the data provided by a Data Source. |
| Attribute | priceDocumentType | PriceDocumentTypeC ode | 0..1 | A code assigned by the Information Provider to indicate to the Party Receiving Private Data, the intended use or purpose of sending the Price Synchronisation Message. The Party Receiving Private Data is able to use this code to determine how to process the information contained within the message. For example, initial load of data, resend of previously sent data or ongoing data synchronisation. |
| <u>Attribute</u> | <u>targetMarketInformationProvid erName</u> | <u>String</u> | <u>0..1</u> | <u>The name of any local provider of price information for an item if this party is different than the Price Synchronisation Document Information Provider.</u> |

4.1.3. Price Synchronisation Confirmation



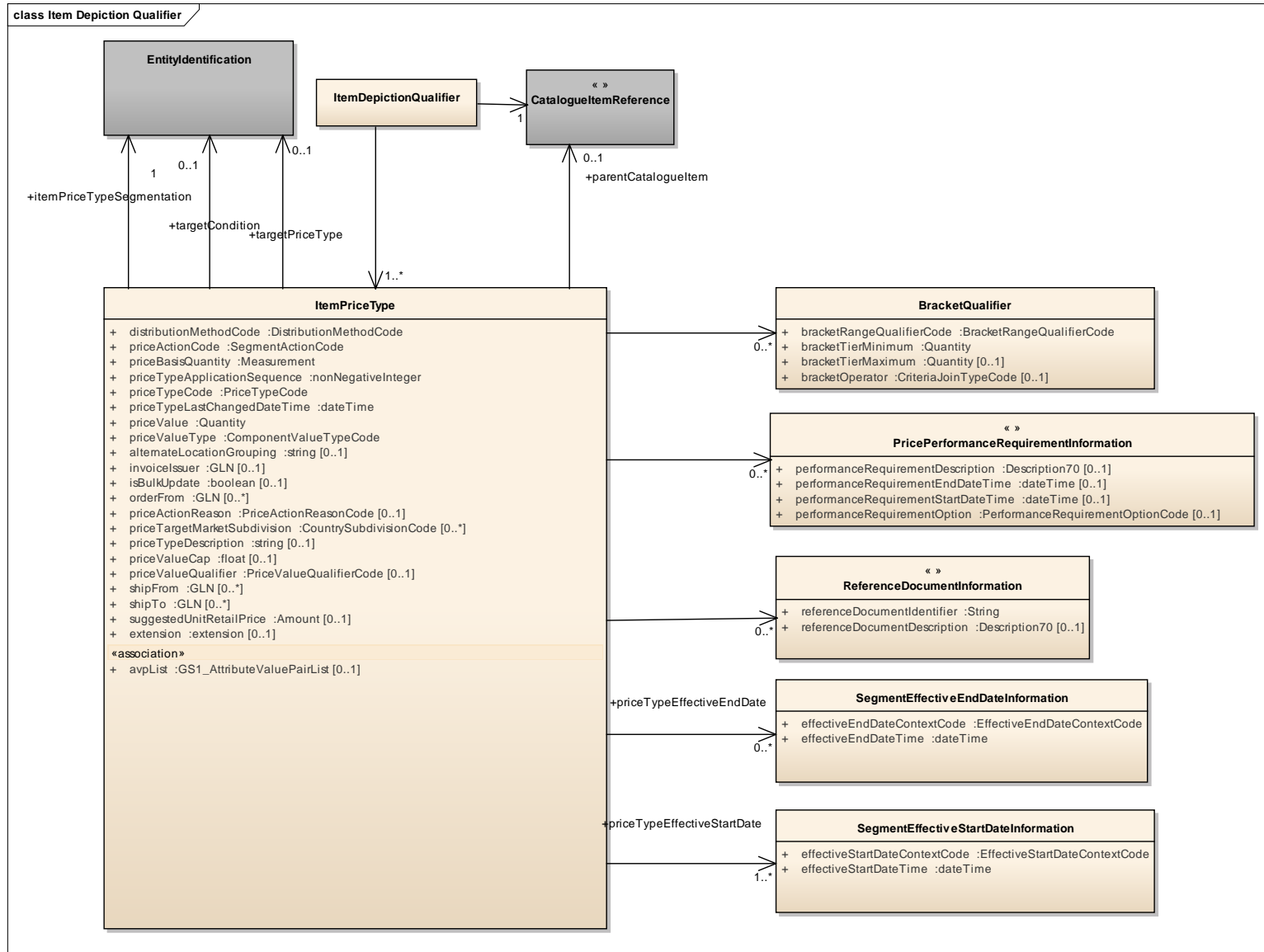
Note: Reference Shared Common Library Business Message (BMS) Release 3.1.0

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|--|------------------|---------------------------|------------------|--|
| PriceSynchronisationSegmentConfirmation | | | | The synchronisation status for the price synchronisation relationship, condition or price segment. |

| Content | Attribute / Role | Datatype /Secondary class | Multi licity | Definition |
|--|--|--|-----------------|---|
| Association | itemPriceTypeSegmentIdentifi cation | EntityIdentification | 1..1 | A string of characters assigned by the Information Provider to uniquely identify a price component associated with an item (from the item price type segment of the price message to which this confirmation is responding). |
| Association | priceSynchronisationRelations hipIdentification | EntityIdentification | 1..1 | A string of characters assigned by the Information Provider to uniquely identify each price synchronization relationship that exists between the Information Provider and the Party Receiving Private Data. Each Price Synchronisation Message can only contain price information related to a single price synchronization relationship (from the relationship segment of the price message to which this confirmation is responding). |
| Association | | Pricesynchronisation ConfirmationStatusReason | 0..* | Provides the reason, action required and relevant attributes and values connected with a synchronisation status. |
| Association | priceSynchronisationConditionI nformation | EntityIdentification | 1..1 | A string of characters assigned by the Information Provider to uniquely identify a summary condition or an item condition of type bracket (from the condition segment of the price message to which this confirmation is responding). |
| Attribute | priceSynchronisationConfirmat ionStatus | SynchronisationConfir mationStatusEnumer ation | 1..1 | Describes the data recipient's action taken on the information contained in a specific segment of the price synchronization message. |
| PricesynchronisationCon firmationStatusReason | | | | Provides further details regarding the synchronisation status for the price synchronisation relationship, condition or price segment including the reason for the status, the action needed and any specific attribute. |
| Attribute | actionNeeded | string | 1..1 | Identifies the type of action the data source needs to take in order to resolve the data recipient's issue. |
| Attribute | confirmationStatusReasonCod e | string | 1..1 | Identifies the type issue the data recipient has with the value communicated in the attribute name. |
| Attribute | priceAttributeName | string | 0..1 | Name of the attribute in the Price Synchronization message. |
| Attribute | priceAttributeValue | string | 0..1 | Value sent in the price synchronisation message that is associated with the attribute name. |

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|--------------------------------------|--|---|------------------|---|
| PriceSynchronisationConfir mation | | | | The electronic communication from the Data Recipient to the Data Source indicating what action has been taken on the price synchronisation relationship, condition or price segment. |
| Generalization | | Document | | |
| Association | priceSynchronisationRelations hipIdentification | EntityIdentification | 1..1 | A string of characters assigned by the Information Provider to uniquely identify each price synchronization relationship that exists between the Information Provider and the Party Receiving Private Data. Each Price Synchronisation Message can only contain price information related to a single price synchronization relationship (from the price synchronization header of the price message to which this confirmation is responding). |
| Association | priceSynchronisationConfirmat ionIdentification | EntityIdentification | 1 ..1 | Uniquely identifies the Price Synchronisation Confirmation |
| Association | priceSynchronisationDocumen tIdentification | EntityIdentification | 1..1 | Within a given price synchronization relationship, a number assigned by the Source Data Pool to uniquely identify each instance of a Price Synchronization Message sent from the Source Data Pool to the Party Receiving Private Data. The number is unique within each Price synchronization relationship (from the price synchronization header of the price message to which this confirmation is responding). |
| Association | | PriceSynchronisation SegmentConfirmation | 1..* | Provides the confirmation status and the applicable price synchronisation segment. |
| Attribute | dataRecipient | GLN | 1..1 | The party receiving the private data (for example, retailer). This information is taken from the price synchronization header of the price message to which this confirmation is responding. |
| Attribute | dataSource | GLN | 1..1 | The information provider of the Price Synchronization message (for example, supplier). This is taken from the price synchronization header of the price message to which this confirmation is responding. |

4.1.4. Item Depiction Qualifier





Note: Reference Shared Common Library Business Message (BMS) Release 3.1.0

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|--|---------------------------------|---------------------------------------|------------------|--|
| ItemDepictionQualifier | | | | A price synchronisation message segment used to show how the pricing information would be depicted on an invoice. |
| Association | | ItemPriceType | 1..* | Associates one or many item price types with an item depiction qualifier. |
| Association | | CatalogueItemReference | 1..1 | Associates a price type with a catalogue item. |
| PriceCommentaryInformation | | | | Details on a price including price value, price value type, price type and description used to depict addition expressions of a price for example list price with VAT |
| Attribute | priceValue | Quantity | 1..1 | Price value used to provide more detail for a parent price.(e.g. \$12.00) |
| Attribute | priceValueType | ComponentValueTy eCode | 1..1 | A classification of the price component used to determine how to apply the amount for example value, rate or percent for a price commentary. |
| Attribute | priceTypeCode | PriceType | 1..1 | A code assigned to identify the kind or class of a price for a descriptive price. |
| Attribute | priceTypeDescription | string | 0..1 | Text used to provide an additional description of a commentary price. |
| BracketQualifier | | | | Identifies conditions required to be met to qualify for a bracket |
| Attribute | bracketRangeQualifierCode | BracketRangeQualifie rCode | 1..1 | Specifies whether the bracket range is based upon an amount, a measurement or another quantity |
| Attribute | bracketTierMinimum | Quantity | 1..1 | The lower limit for qualification for a bracket. The lower limit for qualification for a bracket. |
| Attribute | bracketTierMaximum | Quantity | 0..1 | The upper limits for qualification for a bracket. |
| Attribute | bracketOperator | BracketOperatorCode | 0..1 | A function to identify the logical relationship between multiple bracket qualifiers (And/Or). |
| SegmentEffectiveEndDateI nformation | | | | The effective end date and associated context (e.g. last order date) for a condition. |

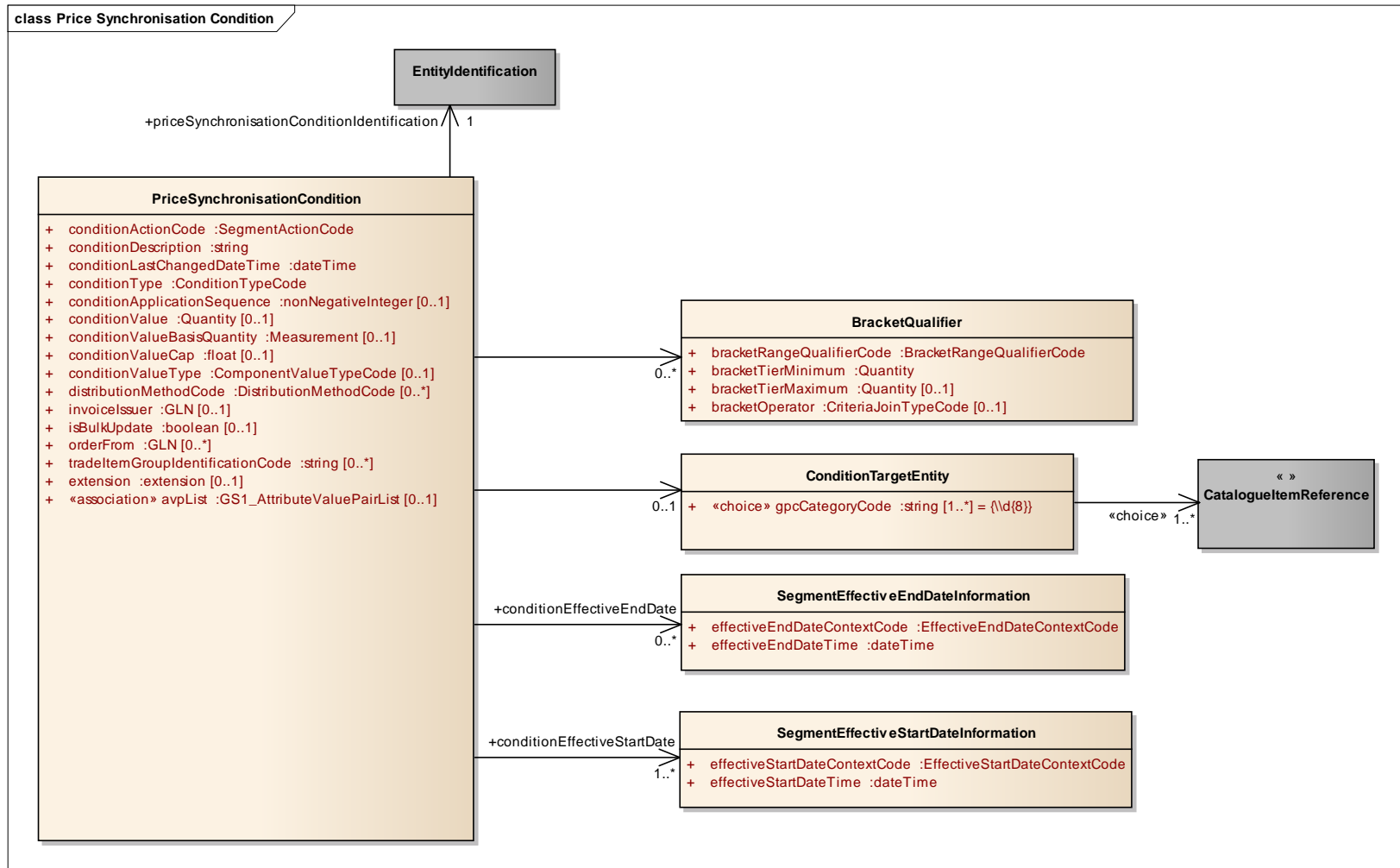
| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|--------------------------------------|-------------------------------|--|------------------|---|
| Attribute | effectiveEndDateTime | dateTime | 1..1 | A date\time used to indicate when the component depicted in the segment is no longer available for use. |
| Attribute | effectiveEndDateContextCode | EffectiveEndDateContextCode | 1..1 | An associated event which gives significance to the effective start date for a segment for example first order date. |
| SegmentEffectiveStartDateInformation | | | | The start date and applicable context for the start date (first order date) for a condition type. |
| Attribute | effectiveStartDateTime | dateTime | 1..1 | The date on which the price synchronisation component begins. |
| Attribute | effectiveStartDateContextCode | EffectiveStartDateContextCode | 1..1 | An associated event which gives significance to the effective start date for a segment for example first order date. |
| ItemPriceType | | | | Contains details of a price component associated with an item. |
| Association | priceTypeEffectiveStartDate | SegmentEffectiveStartDateInformation | 1..* | Provides details on start dates for a given price type. |
| Association | priceTypeEffectiveEndDate | SegmentEffectiveEndDateInformation | 0..* | Provides end date details for a given price type. |
| Association | | PriceCommentaryInformation | 0..* | Details for a price type, on a price including price value, price value type, price type and description used to depict addition expressions of a price for example list price with VAT. — MP01 |
| Association | | PricePerformanceRequirementInformation | 0..* | Provides performance requirements for a price type. |
| Association | | BracketQualifier | 0..* | Provides qualifiers required for eligibility for a price type of bracket. |
| Association | | ReferenceDocumentInformation | 0..* | Provides reference information related to a given price for example a contract number |
| Association | parentCatalogueItem | CatalogueItemReference | 0..1 | A reference to another trade item that is higher in the hierarchal configuration than the item referenced in the Item depiction. Used to vary the price of an item based on a higher level component in a hierarchal configuration. |
| Association | targetCondition | EntityIdentification | 0..1 | A reference to a previous Condition Identification that was used to define the Bracket Qualifiers - |

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|-------------|----------------------------------|--------------------------------|------------------|---|
| | | | | references back to the summary condition Identification. |
| Association | targetPriceType | EntityIdentification | 0..1 | A reference to a previous Price Type Identification that was used to define a component that this price is associated with. |
| Association | avpList | GS1_AttributeValueP airList | 0..1 | Attribute value pair information. |
| Association | itemPriceTypeSegmentation | EntityIdentification | 1..1 | A string of characters assigned by the Information Provider to uniquely identify a price component associated with an item. |
| Attribute | distributionMethodCode | DistributionMethodCo de | 1..1 | The mode by which the Information Provider and the Party Receiving Private Data have agreed at what point(s) in the supply chain the Information Provider makes the goods available to the Party Receiving Private Data. |
| Attribute | priceActionCode | SegmentActionCode | 1..1 | A code assigned by the Information Provider to indicate to the Party Receiving Private Data, the reason for sending the price information contained within the specified segment within the Price Synchronization Message. The Party Receiving Private Data is able to use this code to determine the nature of the action associated with each price component within each price type segment. For example the addition of a new record, the modification of an existing record or the correction of an existing record. |
| Attribute | priceBasisQuantity | Measurement | 1..1 | Price Basis Quantity qualifies Price with a 'Price Per' quantity. This must include a unit of measure to describe what the price and price quantity applies to, such as, a price of \$100 could apply to 1 case of product or to 25 Kilos. Price Basis Quantity includes a Unit of Measure. |
| Attribute | priceTypeApplicationSequenc e | nonNegativeInteger | 1..1 | The order in which the value associated with a price type is applied in the process of calculating the net invoice price. |
| Attribute | priceTypeCode | PriceTypeCode | 1..1 | A code assigned to identify the kind or class of a price component. |
| Attribute | priceTypeLastChangedDateTi me | dateTime | 1..1 | Identifies a certain point in time where the segment was last modified. |

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|-----------|------------------------------|-----------------------------|------------------|---|
| Attribute | priceValue | Quantity | 1..1 | Associates a percent or integer value with a price value. |
| Attribute | priceValueType | ComponentValueTy eCode | 1..1 | A classification of the price component used to determine how to apply the amount for example value, rate or percent. |
| Attribute | alternateLocationGrouping | string | 0..1 | A string of characters used to describe a cluster of business locations mutually defined by the Information Provider and the Party Receiving Private Data. |
| Attribute | extension | extension | 0..1 | |
| Attribute | isBulkUpdate | boolean | 0..1 | Indicates that the update to the price type is for a full price list update. This update may include price increase, price decrease, unchanged prices or new prices. This price change is managed globally by the retailer. |
| Attribute | invoiceIssuer | GLN | 0..1 | The party who issues the invoice for the trade item according to the condition synchronised. |
| Attribute | orderFrom | GLN | 0..* | The location that the item can be ordered from according to the price synchronised |
| Attribute | priceValueCap | float | 0..1 | A quantity or measurement associated with the price value qualifier to limit the calculation of rate to a specified maximum amount. This would be used where a trading partner sets a maximum value for an offer. |
| Attribute | priceValueQualifier | PriceValueQualifierC ode | 0..1 | A code assigned to identify the basis on which a specific price value is acted upon. For example, if the Price Value was 2%, the Price Value Qualifier would be 'percent'. |
| Attribute | priceActionReason | PriceActionreasonCo de | 0..1 | A code to indicate the justification or explanation as to why the action associated with each price component has occurred. All actions may have an associated reason. |
| Attribute | priceTargetMarketSubdivision | CountrySubdivisionC ode | 0..* | The code for country sub-division used to indicate the geo-political subdivision of the target market (=country). |
| Attribute | priceTypeDescription | string | 0..1 | Text used to provide an additional description of the price component. |

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|---|---|--------------------------------------|----------------------|---|
| Attribute | shipFrom | GLN | 0..* | Identifies the origin location from where the goods will be shipped. |
| Attribute | shipTo | GLN | 0..* | The location destination to which goods will be shipped. |
| Attribute | suggestedUnitRetailPrice | Amount | 0..1 | The retail (to consumer) price as suggested by the manufacturer. This is normally used to establish a proposed value for the trade item for marketing purposes. May or may not appear on the package. |
| PricePerformanceRequirem entInformation | | | | Standardized list of requirements which are types of price components to be met to receive a monetary value. |
| Attribute | performanceRequirementDesc ription | Description70 | 0..1 | A string of characters used to describe additional or more specific requirements to be met in order to receive a monetary value. |
| Attribute | performanceRequirementEnd DateTime | dateTime | 1..1 | A date indicating the ending of a period during which the performance requirements should be met. |
| Attribute | performanceRequirementStart DateTime | dateTime | 1..1 | A date indicating the beginning of a period during which the performance requirements should be met. |
| Attribute | performanceRequirementOptio n | performanceRequire mentOptionCode | 0..1 | Standardized list of requirements which are types of price components to be met to receive a monetary value. |
| ReferenceDocumentInform ation | | | | This class enables the input of a reference document (e.g. contract) for a specific condition. |
| Attribute | referenceDocumentIdentifier | string | 1..1 | Identifier that provides a link to further detail on the price condition, for example an associated contract between trading partners. |
| Attribute | referenceDocumentDescriptio n | Description70 | 0..1 | A free form text field used to describe a contract or other document which contains more information about agreements made regarding a condition. |

4.1.5. Price Synchronisation Condition



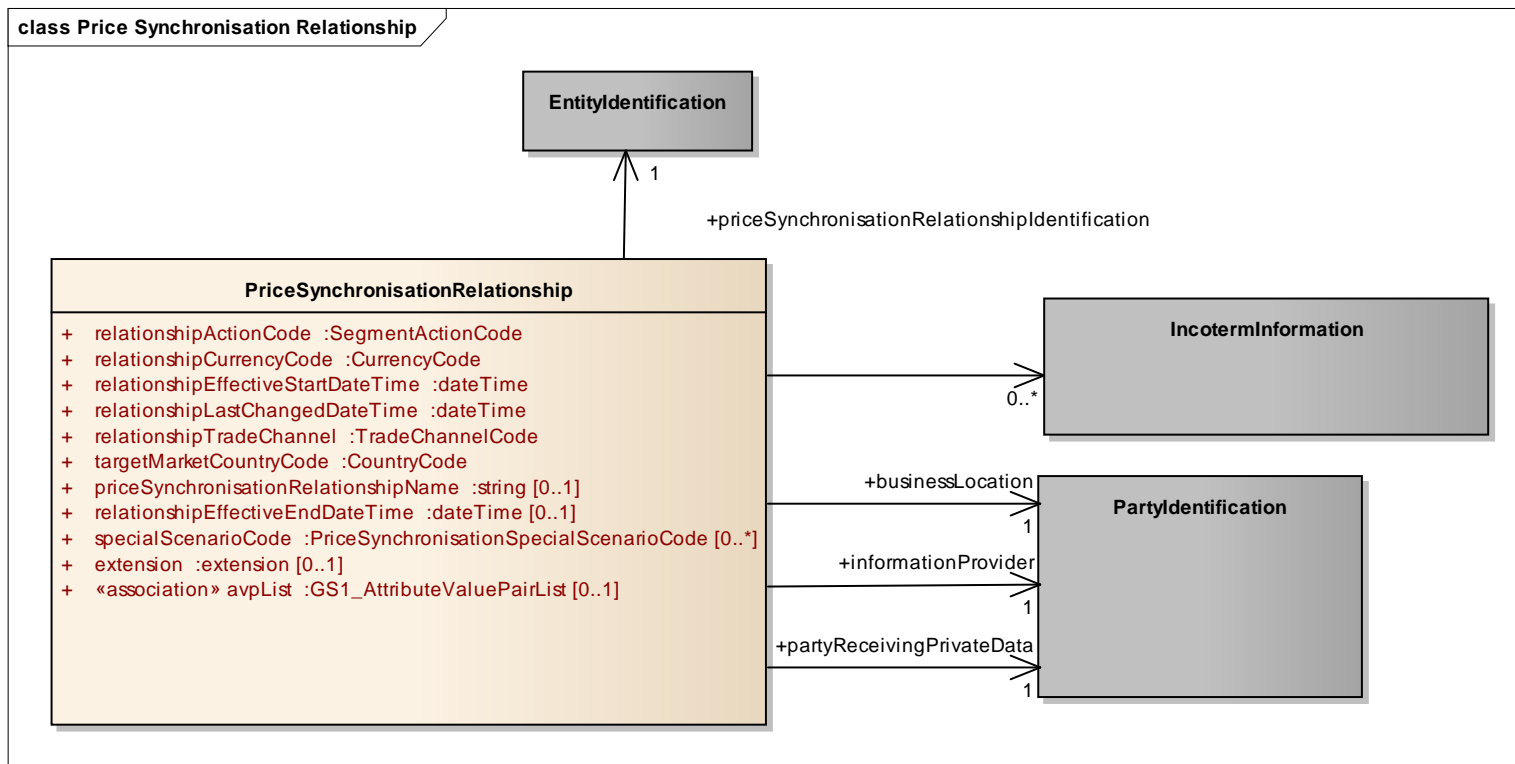
Note: Reference Shared Common Library Business Message (BMS) Release 3.1.0

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|---|---|--|------------------|--|
| PriceSynchronisationCon dition | | | | |
| Association | | ConditionTargetEntity | 0..1 | Provides an item of grouping of items associated with a price condition |
| Association | priceSynchronisationCondition Identification | EntityIdentification | 1..1 | A string of characters assigned by the Information Provider to uniquely identify a summary condition or an item condition of type bracket. |
| Association | | BracketQualifier | 0..* | Provides conditions for being qualified for a given bracket. |
| Association | conditionEffectiveStartDate | SegmentEffectiveStar tDateInformation | 1..* | Provides the effective start date and context for a price synchronisation condition. |
| Association | conditionEffectiveEndDate | SegmentEffectiveEnd DateInformation | 0..* | Provides the effective end date and context for a price synchronisation condition |
| Association | avpList | GS1_AttributeValueP airList | 0..1 | Attribute value pair information. |
| Attribute | conditionActionCode | SegmentActionCode | 1..1 | A code assigned by the Information Provider to indicate to the Party Receiving Private Data, the reason for sending the price information contained within the specified segment within the Price Synchronization Message. The Party Receiving Private Data is able to use this code to determine the nature of the action associated with each condition within each condition segment. For example the addition of a new record, the modification of an existing record or the correction of an existing record. |
| Attribute | conditionDescription | string | 1..1 | Text used to provide an additional description of the condition |
| Attribute | conditionLastChangedDateTim e | dateTime | 1..1 | Identifies a certain point in time where the segment was last modified. |
| Attribute | conditionType | ConditionTypeCode | 1..1 | Condition types are general classifications for a given condition. The treatment of the values in a price calculation is determined by the Condition Type. |
| Attribute | conditionApplicationSequence | nonNegativeInteger | 0..1 | The order in which the value associated with a summary condition type of allowance or charge, is applied in the process of calculating the net invoice |

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|-----------------------|--------------------------------------|----------------------------|------------------|---|
| | | | | price. |
| Attribute | conditionValue | Quantity | 0..1 | Provides a value or percent associated with a condition |
| Attribute | conditionValueType | ComponentValueTyp eCode | 0..1 | A classification of the price component used to determine how to apply the amount for example value, rate or percent. |
| Attribute | conditionValueCap | float | 0..1 | A quantity or measurement associated with the condition value qualifier to limit the calculation of rate to a specified maximum amount. This would be used where a trading partner sets a maximum value for an offer. |
| Attribute | conditionValueBasisQuantity | Measurement | 0..1 | The base amount used for a condition in the case or a rate for example \$10 per '100' yards where 100 yards is the value basis. |
| Attribute | distributionMethodCode | DistributionMethodCo de | 0..* | The mode by which the Information Provider and the Party Receiving Private Data have agreed at what point(s) in the supply chain the Information Provider makes the goods available to the Party Receiving Private Data. |
| Attribute | extension | extension | 0..1 | |
| Attribute | invoiceIssuer | GLN | 0..1 | The party who issues the invoice for the trade item according to the condition synchronised. |
| Attribute | isBulkUpdate | boolean | 0..1 | Indicates that the update to the condition is for a full price list update. This update may include price increase, price decrease, unchanged prices or new prices. This price change is managed globally by the retailer |
| Attribute | orderFrom | GLN | 0..* | The location that the item can be ordered from according to the condition synchronised. |
| Attribute | tradeItemGroupIdentificationC ode | string | 0..* | A code assigned by the supplier or manufacturer to logically group trade item independently from the Global trade item Classification. |
| ConditionTargetEntity | | | | Provides the specific item or groups of items that the condition applies to |
| Association | <<Choice>> | CatalogueItemRefere nce | 1..* | Associates one or many catalogue items with a price type. |

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|--|-----------------------------------|-----------------------------------|------------------|--|
| Attribute | <<Choice>>gpcCategoryCode | string | 1..* | The GS1 provided code which identifies the Global Product Classification Attribute Value. |
| SegmentEffectiveEndDateI nformation | | | | The effective end date and associated context (e.g. last order date) for a condition. |
| Attribute | effectiveEndDateContextCode | EffectiveEndDatecont extCode | 1..1 | An associated event which gives significance to the effective start date for a segment for example first order date. |
| Attribute | effectiveEndDateTime | dateTime | 1..1 | A date\time used to indicate when the component depicted in the segment is no longer available for use. |
| SegmentEffectiveStartDateI nformation | | | | The start date and applicable context for the start date (first order date) for a condition type. |
| Attribute | effectiveStartDateContextCod e | EffectiveStartDateCo ntextCode | 1..1 | An associated event which gives significance to the effective start date for a segment for example first order date. |
| Attribute | effectiveStartDateTime | dateTime | 1..1 | The date on which the price synchronisation component begins. |

4.1.6. Price Synchronisation Relationship



Note: Reference Shared Common Library Business Message (BMS) Release 3.1.0

| Content | Attribute / Role | Datatype /Secondary class | Multiplicity | Definition |
|----------------------------------|--|---------------------------|--------------|--|
| PriceSynchronisationRelationship | | | | A message segment used to establish a price synchronisation relationship between trading partners. |
| Association | priceSynchronisationRelationshipIdentification | EntityIdentification | 1..1 | Identifies a unique buyer-seller price sync relationship generated by the data source. |
| Association | informationProvider | PartyIdentification | 1..1 | The party who owns the data |
| Association | partyReceivingPrivateData | PartyIdentification | 1..1 | Party, which is authorized to view, use, download |

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|-------------|--|----------------------------|------------------|---|
| | | | | the data provided by a Data Source. |
| Association | businessLocation | PartyIdentification | 1..1 | An entity that belongs to the Party Receiving Private Data, who is the intended recipient of the price information contained within the Price Synchronization Message. |
| Association | | IncotermInformation | 0..* | Provides incoterm details applicable to a trading partner relationship. |
| Association | avpList | GS1_AttributeValuePairList | 0..1 | Attribute value pair information. |
| Attribute | relationshipActionCode | SegmentActionCode | 1..1 | Indicates how the trading partner applies the information in the specified segment |
| Attribute | relationshipCurrencyCode | CurrencyCode | 1..1 | A code used to indicate the system of money used within a particular country by the trading partners to conduct their commercial transactions. |
| Attribute | relationshipEffectiveStartDateTime | dateTime | 1..1 | The day on which the price synchronization relationship commences. |
| Attribute | relationshipLastChangedDateT ime | dateTime | 1..1 | Identifies a certain point in time where the segment was last modified. |
| Attribute | relationshipTradeChannel | TradeChannelCode | 1..1 | Used to specify how the trading partners within a price synchronization relation agree to define the distribution or marketing segmentation of products, customers and geographic areas into common groups that are supplied, serviced and measured in similar ways. The Trade Channel may be defined in the context of the Party Receiving Private Data. |
| Attribute | targetMarketCountryCode | CountryCode | 1..1 | The target market code indicates the country level or higher geographical definition in which the price information is applicable. |
| Attribute | extension | extension | 0..1 | An extension point for a Price Synchronisation Relationship. |
| Attribute | priceSynchronisationRelations hipName | string | 0..1 | The name assigned by the buyer and seller to their price sync relationship. |
| Attribute | relationshipEffectiveEndDateTi me | dateTime | 0..1 | The day on which the price synchronization relationship ends. |
| Attribute | specialScenarioCode | PriceSynchronisation | 0..* | A specialised price synchronisation scenario that may be prevalent in a certain target market based |

| Content | Attribute / Role | Datatype /Secondary class | Multipl icity | Definition |
|---------|------------------|---------------------------|------------------|--|
| | | SpecialScenarioCode | | on regional business practices or regulations. This attribute is used to trigger special processing by the data source and/or data recipient based on the needs of this scenario. This attribute uses the PriceSynchronisationSpecialScenarioCode. |

4.2. Code Lists



Note: Reference Shared Common Library Business Message (BMS) Release 3.0.0 and GDSN Domain Common Library Business Message (BMS) Release for all common code lists.

4.2.1. Bracket Range Qualifier Code

| | |
|-----------------------|---------------------------|
| GS1 Code List | BracketRangeQualifierCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| Code List Name | Code List Description |
|-------------------|---------------------------------------|
| AMOUNT_RANGE | A range value with a currency. |
| MEASUREMENT_RANGE | A range value using a Unit Of Measure |
| RANGE | A numeric range. |

4.2.2. Component Value Type Code

| | |
|-----------------------|------------------------|
| GS1 Code List | ComponentValueTypeCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| Code List Name | Code List Description |
|----------------|---|
| PERCENT | A part of a whole expressed in hundredths |
| VALUE | A numerical quantity that is assigned or is determined by calculation or measurement. |

4.2.3. Condition Type Code

| | |
|------------------------------|--------------------------|
| GS1 Code List | ConditionTypeCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| Code List Name | Code List Description |
|------------------------------|---|
| ALLOWANCE | Credit reflected on an invoice. This can occur either at the item or the invoice level. There are many types of allowances. Some are contractually based, e.g. backhaul, others are not formalized contracts. Some allowances are offered to the industry such as payment terms while others such as promotional allowances are trading partner specific. |
| BRACKET | The price associated with an item for the purchase of a specific number of trade items, or some other logistical measure (weight, cube, truck). These are often offered in a series (e.g. 100 to 299 case lots, 300 to 599, full truckload, half truckload; each offering a different discount). |
| CHARGE | Debit reflected on an invoice. This can occur either at the item or at the invoice level. There are many types of charges. Some are contractually based, others are not formalized contracts. |
| PRICE_NOTIFICATION_LEAD_TIME | Number of calendar days from a stated effective date that a price becomes valid for application. |
| ROUNDING_FACTOR | The number of positions to the right of the decimal (or comma) that trading partners agree to define as the precision of the numerical value communicated between the partners. |

4.2.4. Effective End Date Context Code

| | |
|------------------------------|------------------------------------|
| GS1 Code List | EffectiveEndDateContextCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| Code List Name | Code List Description |
|--------------------|--|
| AD_END_DATE | The end date for an advertisement for a given product. |
| LAST_DELIVERY_DATE | The day on which the last delivery is made. |
| LAST_ORDER_DATE | It indicates the latest date that an order can be placed for the trade item. |
| LAST_SHIP_DATE | It indicates the latest date that the trade item can be shipped. This is independent of any specific ship-from location. |

4.2.5. Effective Start Date Context Code

| | |
|------------------------------|--------------------------------------|
| GS1 Code List | EffectiveStartDateContextCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| Code List Name | Code List Description |
|---------------------|--|
| AD_START_DATE | The start date for an advertisement for a given product. |
| FIRST_DELIVERY_DATE | The day on which the first delivery is made. Also know as First Arrival Date. |
| FIRST_ORDER_DATE | It indicates the earliest date that an order can be placed for the trade item. |
| FIRST_SHIP_DATE | It indicates the earliest date that the trade item can be shipped. This is independent of any specific ship-from location. |

4.2.6. Performance Requirement Option Code

| | |
|------------------------------|---|
| GS1 Code List | PerformanceRequirementOptionCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| Code List Name | Code List Description |
|----------------------|-----------------------|
| AISLE_DISPLAY | Not Available |
| BILLBOARD_AD | Not Available |
| BOGO | Not Available |
| CHART | Not Available |
| COUPON | Not Available |
| DIRECT_MAIL | Not Available |
| DUMP_BIN | Not Available |
| END_CAP_DISPLAY | Not Available |
| FLOOR_GRAPHICS | Not Available |
| FLOOR_STACK | Not Available |
| FLYER | Not Available |
| FREE_ITEM | Not Available |
| GONDOLA_DISPLAY | Not Available |
| IN_STORE_SPECIAL | Not Available |
| IN_STORE_DEMO_SAMPLE | Not Available |

| | |
|------------------------|----------------------|
| IN_STORE_DISPENSER | Not Available |
| INSERT | Not Available |
| INSTANT_REBATE | Not Available |
| INTERNET_AD | Not Available |
| ITEM_INTRO | Not Available |
| MAIL_IN_REBATE | Not Available |
| MEMBERSHIP_CARD | Not Available |
| NEWSPAPER_AD | Not Available |
| ON_COUNTER | Not Available |
| OTHER | An unspecified type. |
| PERIMETER_DISPLAY | Not Available |
| PURCHASE_WITH_PURCHASE | Not Available |
| RACK_DISPLAY | Not Available |
| RADIO_AD | Not Available |
| RETAILER_CIRCULAR | Not Available |
| SCANNER | Not Available |
| SHELF_EXTENDER | Not Available |
| SHIPPER_DISPLAY | Not Available |

4.2.7. Price Action Reason Code

| | |
|------------------------------|------------------------------|
| GS1 Code List | PriceActionReasonCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| Code List Name | Code List Description |
|----------------|---------------------------------|
| NI | The introduction of a new item. |
| PD | Price decrease. |
| PI | A price increase. |
| RE | Range extension. |
| SC | Size change (Pack or Pallet). |
| TPR | Temporary price reduction. |

4.2.8. Price Document Type Code

| | |
|------------------------------|------------------------------|
| GS1 Code List | PriceDocumentTypeCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| Code List Name | Code List Description |
|----------------|--|
| INITIAL_LOAD | The sending of pricing information for the first time. |
| RELOAD | Sending all current and known future pricing. This is used to start over by replacing previously synchronised information. |
| RESEND | Indicates that the message is used to recover a lost or missing message. |
| RESTART | The status used when a data recipient had rejected an item's pricing but wishes to resume price synchronisation. |

4.2.9. Price Synchronisation Special Scenario Code

| | |
|------------------------------|--|
| GS1 Code List | PriceSynchronisationSpecialScenarioCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| | |
|---|---|
| 1 | Resynchronisation of all Price Types for pricing done at the lowest level consumer unit. Uses NO_ACTION code to differentiate price types that have changed from those that have not. |
|---|---|

4.2.10. Price Type Code

| | |
|------------------------------|----------------------|
| GS1 Code List | PriceTypeCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| | |
|------------------------|---|
| ALLOWANCE | Credit reflected on an invoice. This can occur either at the item or the invoice level. There are many types of allowances. Some are contractually based, e.g. backhaul, others are not formalized contracts. Some allowances are offered to the industry such as payment terms while others such as promotional allowances are trading partner specific. |
| BRACKET_TIER_PRICE | The price associated with an item for the purchase of a specific number of trade items, or some other logistical measure (weight, cube, truck). These are often offered in a series (e.g. 100 to 299 case lots, 300 to 599, full truckload, half truckload; each offering a different discount). |
| CHARGE | Debit reflected on an invoice. This can occur either at the item or at the invoice level. There are many types of charges. Some are contractually based, others are not formalized contracts. |
| CONTRACT_PRICE | The price associated with an item that has been negotiated or agreed to between trading partners exclusive of taxes but inclusive of allowances, charges and customs duty where applicable. This price would typically be associated with a formal contract between trading partners. |
| DECLARED_CUSTOMS_VALUE | This is the value of the item as declared by the supplier. This value is used by customs to calculate the customs duty that is payable. |
| INTRODUCTORY_PRICE | Not Available |
| LIST_PRICE | External price associated with a product absent of all allowances or charges. This is normally the printed price contained on supplier's price list or catalogue. (May or may not be customer specific). |
| OTHER | An unspecified type. |
| PROMOTIONAL_PRICE | A price that is available only as part of a promotion. |
| RETAIL_PRICE | The retail (to consumer) price as suggested by the manufacturer. May or may not appear on the package. |

| | |
|--|--|
| TRANSACTION_PRICE | The line item/GTIN price shown on the invoice document, including allowances and/or charges applying to the trading partner relationship (which can be zero), but excluding VAT and any other taxes, fees, and/or duties. |
| TRANSACTION_PRICE_WITH_SPECIAL_TAXES | The line item / GTIN price shown on the invoice document, including allowances and/ or charges applying to the trading partner relationship (which can be zero), excluding VAT, but including any other taxes, fees, and/or duties. |
| TRANSACTION_PRICE_WITH_VAT_AND_SPECIAL_TAXES | The line item / GTIN price shown on the invoice document, including allowances and/ or charges applying to the trading partner relationship (which can be zero), including both VAT and any other taxes, fees, and/or duties. |
| UNDERBOND_LIST_PRICE | The price associated with an item exclusive of all allowances, charges, taxes and customs duty. (May or may not be customer specific). This applies to imported items that upon arrival into the country are stored in a Customs bonded warehouse. The goods are then sold without the customs duty paid. The purchaser must pay the customs duty. |
| UNDERBOND_TRANSACTION_PRICE | The price associated with an item exclusive of all taxes and customs duty but inclusive of all allowances and charges. This applies to imported items that upon arrival into the country are stored in a Customs bonded warehouse. The goods are then sold without the customs duty paid. The purchaser must pay the customs duty. |

4.2.11. Price Value Qualifier Code

| | |
|------------------------------|--------------------------------|
| GS1 Code List | PriceValueQualifierCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| Code List Name | Code List Description |
|-----------------|------------------------------------|
| MONETARY_AMOUNT | An amount of or relating to money. |
| PERCENT | One part in a hundred. |

4.2.12. Segment Action Code

| | |
|------------------------------|--------------------------|
| GS1 Code List | SegmentActionCode |
| GS1 Code List Version | R1 |
| Managing Agency | GS1 |
| Based on Code List | n/a |
| Type Of Management | n/a |

| Code List Name | Code List Description |
|-------------------|---|
| ADD | Used to signify that a Data Source is seeking to synchronise new data with the Data Recipient |
| CORRECT | Used to error correct or change the values of mandatory key attributes or an attribute where the change results in material financial impact. |
| DELETE | Used to remove one or many iterations of an existing segment. |
| CHANGE_BY_REFRESH | Used to change the values of any of the optional attributes within the segment. |
| NO_ACTION | No change or correction is being made to the segment. |

5. Business Document Example

| Attribute | Value |
|--|------------------|
| PriceSynchronisationDocument | |
| Document | |
| creationDateTime | 2011-03-11 11:00 |
| documentStatus | ORIGINAL |
| <ul style="list-style-type: none"> informationProvider | 0012345000010 |
| <ul style="list-style-type: none"> partyReceivingPrivateData | 0056345000022 |
| <ul style="list-style-type: none"> priceDocumentType | INITIAL_LOAD |
| priceSynchronisationDocumentationIdentification | |
| - entityIdentification | 20051101 |
| PartyIdentification (contentOwner) | |
| - gln | 8712345678913 |
| priceSynchronisationRelationshipIdentification | |
| - entityIdentification | 20051102 |
| PartyIdentification (contentOwner) | |
| - gln | 8712345678913 |
| PriceSynchronisationRelationship | |
| <ul style="list-style-type: none"> PriceSynchronisationRelationshipIdentification | |
| - entityIdentification | 20051101 |

| Attribute | Value |
|--|-------------------------------|
| PartyIdentification (contentOwner) | |
| - gln | 8712345678913 |
| • priceSynchronisationRelationshipName | Nana Corporation Food Service |
| • relationshipAction | ADD |
| • relationshipCurrencyCode | USD |
| • relationshipEffectiveEndDateTime | 2007-01-10T12:00:01.000 |
| • relationshipEffectiveStartDateTime | 2006-01-10T12:00:01.000 |
| • relationshipLastChangedDateTime | 2007-01-10T12:00:01.000 |
| • relationshipTradeChannel | FOOD_SERVICE |
| • targetMarketCountryCode | US |
| • informationProvider | 0012345000010 |
| • businessLocation | 0012345000010 |
| • partyReceivingPrivateData | 0056345000022 |
| IncotermInformation | |
| • incotermCode | CFR |
| • incotermCodeLocation | Port Charlotte |
| PriceSynchronisationCondition | |
| • conditionActionCode | ADD |
| • conditionApplicationSequence | 1 |
| • conditionDescription | Extremely Large Order Bracket |
| • conditionLastChangedDateTime | 2007-01-10T12:00:01.000 |
| • conditionType | BRACKET |
| • conditionValueBasisQuantity | 10,000 YD |
| • conditionValue | 10.00 |
| • conditionValueType | PERCENT |
| • conditionValueCap | 10.00 |
| PriceSynchronisationConditionIdentification | |
| ○ uniqueCreatorIdentification | WG-000007 |
| ○ contentOwner | 0012345000010 |
| conditionEffectiveStartDate | |
| ○ effectiveStartDateTime | 2006-01-10T12:00:01.000 |
| ○ effectiveStartDateContextCode | FIRST_DELIVERY_DATE |
| conditionEffectiveEndDate | |
| ○ effectiveEndDateTime | 2007-01-10T12:00:01.000 |
| ○ effectiveEndDateContextCode | LAST_DELIVERY_DATE |
| BracketQualifier | |
| • bracketRangeQualifierCode | MEASUREMENT_RANGE |

| Attribute | Value |
|---|---------------------------|
| • bracketTierMaximum | 500,000 YD |
| • bracketTierMinimum | 100,000 YD |
| ConditionTargetEntity | |
| • CatalogueItemReference | |
| o gtin | 06110123456784 |
| o dataSource | 0012345000010 |
| o targetMarketCountryCode | US |
| ItemDepictionQualifier | |
| ItemPriceType | |
| • alternateLocationGrouping | 72436437 |
| • distributionMethodCode | CD |
| • priceActionCode | ADD |
| • priceActionReason | NI |
| • priceTargetMarketSubdivision | US-CA |
| • priceTypeApplicationSequence | 1 |
| • priceTypeCode | INTRODUCTORY_PRICE |
| • priceTypeLastChangedDateTime | 2007-01-10T12:00:01.000 |
| • shipFrom | 0012345000011 |
| • shipTo | 0056345000025 |
| • suggestedUnitRetailPrice | 30.00 USD |
| • priceBasisQuantity | 100 YD |
| • priceValue | 10.00 |
| • priceValueQualifier | MONETARY_AMOUNT |
| • priceValueType | VALUE |
| priceTypeEffectiveStartDate | |
| • effectiveStartDateTime | 2006-01-10T12:00:01.000 |
| • effectiveStartDateContextCode | FIRST_DELIVERY_DATE |
| ReferenceDocumentationInformation | |
| • referenceDocumentIdentifier | 123232334334 |
| • referenceDocumentDescription | Contract dated 2006-07-01 |
| PricePerformanceRequirementInformation | |
| • performanceRequirementEndDateTime | 2007-01-10T12:00:01.000 |
| • performanceRequirementOption | INSERT |
| • performanceRequirementStartDateTime | 2006-01-10T12:00:01.000 |
| targetPriceType | |
| o uniqueCreatorIdentification | WG-000005 |
| o contentOwner | 0012345000010 |
| itemPriceTypeSegmentation | |

| Attribute | Value |
|------------------------------------|---------------|
| - entityIdentification | 20051101 |
| PartyIdentification (contentOwner) | |
| - gln | 8712345678913 |

| Attribute | Value |
|--|---------------|
| PriceSynchronisationConfirmation | |
| • dataRecipient | 0012345000010 |
| • dataSource | 0056345000022 |
| • priceSynchronisationDocumentIdentification | |
| - entityIdentification | 20051101 |
| PartyIdentification (contentOwner) | |
| - gln | 8712345678913 |
| • priceSynchronisationConfirmationIdentification | |
| - entityIdentification | 20051102 |
| PartyIdentification (contentOwner) | |
| - gln | 8712345678913 |
| • priceSynchronisationRelationshipIdentification | |
| - entityIdentification | 20051103 |
| PartyIdentification (contentOwner) | |
| - gln | 8712345678913 |
| PriceSynchronisationSegmentConfirmation | |
| ○ priceSynchronisationConfirmationStatus | REVIEW |
| • priceSynchronisationRelationshipIdentification | |
| - entityIdentification | 20051103 |
| PartyIdentification (contentOwner) | |
| - gln | 8712345678913 |

6. Implementation Considerations

6.1.1. Bulk Update

Bulk updates are full updates to price lists that may occur annually or multi-annually. The full set of trade items list price is published with a new reference number. This update may include price increase, price decrease, unchanged prices or new prices.

This price change is managed globally by the retailer.

There is a need to group the full list of published prices from a trading partner on a single price synchronisation document to be processed at the same time. When these prices are grouped on one single table, the retailer can perform multiple controls: comparison between the new and the previous list prices, weighting of the global price change, identification of the gaps in the list of trade items between previous and new price lists. A bulk update can serve as a trigger for these activities.

Data recipient activity in relation to bulk updates will be triggered by isBulkUpload flag located in the condition and price segments.

6.1.2. Initial Load

Initial Load is defined as sending any pricing information for the first time for items that have already been communicated between trading partners via GDSN; after that, all pricing information sent through the GDSN will be an Add, CHANGE_BY_REFRESH, Correct, or Delete.

- Initial load can happen by data recipient territory or data source categories.
- There may be multiple initial loads until entire Catalogue is synchronized.

The initial load is not used to signify pricing for a new Catalogue Item

- A “new catalogue item” should be indicated in the Reason for Price Change attribute)
- Price messaging requirements

First Initial Load

- Price Document ID must = 1
- Price Document Type must = “Initial Load”
- All segments must be ADDs
- No dependency checks are performed on the confirmation status codes for any segments

Subsequent Initial Load (for different product types, etc.)

- Price Document ID must be > 1
- All price message segments must be ADDs
- Dependency checks must be performed
- Source Data Pool sends the entire price message to the data recipient
- Subsequent messages must have all confirmation status validation rules applied

If Price document Type = “Resend” the Initial Load validation rules are bypassed.

Relationship Segment

- No Response
 - Can continue Initial Load on any segment
- Received/Synchronized/Review
 - Can take modifications on any segment
 - Depends upon individual segment status

Condition Segment

- No Response stops modifications on this and any price type(s) that refer to this condition
- Can still continue an Initial Load of any other price types

6.1.3. Resend

Resend is to recover a lost or missing message only

- File level request
- Price Document Type = "Resend"
- Source Data Pool will send an exact copy of Price Document ID that is requested
- The original Price document will not have a Price Document ID Type
- Source Data Pool will need to change Price Document ID Type to "Resend".
- No additional dependency checks are performed
- The sync list is updated with the new transmission date.

6.1.4. Reload

Reload is a request to "start over" by sending all current and future pricing

- It is a relationship level request
- Price Document Type = "Reload"
- Synchronisation Header Action Code = ADD
- Relationship segment can be resent as ADD
- All price message segment action codes must be ADD
- Document ID will be 1
- No dependency checks are performed

6.1.5. Restart

RESTART Document Type is used to 'restart' pricing for an item the retailer has previously rejected pricing for. "The RESTART synchronization applies to Price Types; note that condition segments cannot be restarted (since they are not at a single GTIN level) and do not apply to this document type.

This is an item level request submitted by the supplier at the request of the retailer. Restart is relationship specific. Therefore, RESTART does not span across multiple retailers or suppliers. Likewise, if the rejected prices span more than 1 price relationship for the requesting retailer, separate restart requests must be submitted for each item / price relationship.

The supplier is responsible for re-synching all price types (active and future) for the given item and price relationship via the SDP. Given price synchronization has stopped for rejected price types, the SDP will not restart any price types for the specific item and price relationship without the supplier providing full information for each price type to be restarted.

SDP:

1. Applies price type updates which include performing dependency checks for each price type segment received from the supplier as part of the RESTART request.
2. Resets price confirmation responses to initial default value for each price type being restarted. This is the only method a retailer has to reset a 'REJECT' response.
3. Sends price synchronization message containing the price type segments only (does not affect conditions) related to the item being restarted:
 - Price Document Type = 'RESTART'
 - Synchronisation Header Action Code = 'CHANGE_BY_REFRESH'
 - All price message segment action codes = 'ADD'
 - Document ID is next sequential document ID pertaining to this retailer and price relationship (do not restart to 1)

RDP:

1. Receives the RESTART price message containing the restarted price type updates.
2. Resets price confirmation responses to initial default value for each price type being restarted.
3. Forwards RESTART price message onto the targeted recipient.

Recipient:

1. Receives 'RESTART' price message
2. Replaces all existing price types for the item / price relationship with those received. Any old pricing for the item / price relationship not received as part of the RESTART price message should be inactivated to ensure that the recipient only maintains the current state of pricing for this item / price relationship.

6.2. Price Sequencing Rules

6.2.1. Item Price Types

1. All Price Types must have an Application Sequence assigned.
2. All Allowances & Charge Price Types must be calculated before applying any Summary Conditions.
3. The Target Price, referenced in the Allowance or Charge Price Type becomes the starting point for the net invoice calculation.
4. All Price Types, other than Price Types = 'Allowance' or 'Charge', must be assigned an Application Sequence = 1.
5. Price Types = 'Allowance' or 'Charge' must be assigned an Application Sequence >1.
6. If Application Sequence = 2 the calculation is derived from the relevant price with Application Sequence = 1.

7. If Application Sequence is >2 the calculation is derived from the prior subtotal.
8. The same Application Sequence # can be applied to more than one Price Type associated with the item. If this is the case, and the Price Type was either an allowance or a Charge, the allowance or charge would be applied to the same prior subtotal.
9. Application Sequence #'s for price types may not always be in a continuous numerical sequence i.e. there may be missing sequence #'s. For example 1,3,4. Therefore you would simply go to the next highest number in the sequence. However there must be at least one Price Type with an Application Sequence = 1.
10. Price types need to be grouped and applied in numerical sequence starting with Application Sequence = 1.

6.2.2. Summary Conditions

1. Only Condition Type = 'Allowance' or 'Charge' would have an Application Sequence assigned.
2. Application Sequence = 1 is not a valid Application Sequence for Summary Conditions.
3. Summary Conditions can only be applied to the calculation after all Allowance & Charge Price Types have been applied to the calculation of the net invoice price.
4. If Application Sequence = 2 the calculation is derived from the Starting Prices on the invoice.
5. If Application Sequence = 3 the calculation is derived from the item subtotals of the items.
6. If Application Sequence is >3 the calculation is derived from the prior subtotal.
7. The same Application Sequence # can be applied to multiple summary conditions. If this is the case, the conditions would be applied to the same prior item subtotal or subtotal as applicable.
8. Application Sequence #'s for Summary Conditions may not always be in a continuous numerical sequence i.e. there may be missing sequence #'s. For example 1, 3, 4. Therefore you would simply go to the next highest number in the sequence.
9. Summary Conditions need to be grouped and applied in numerical sequence starting with Application Sequence =1.

6.3. Communicating Multiple Catalogue Item Qualifiers

Catalogue Item a

- Price Type: 100 \$10 Bracket 1
- Price Type: 101 \$10 Bracket 1
- Price Type: 102 \$10 Bracket 1
- Price Type: 200 \$10 Bracket 1 Target Price Type=100
- Price Type: 201 \$10 Bracket 1 Target Price Type=102
- Price Type: 202 \$10 Bracket 1 (not populated = all items)

Catalogue Item b

- Price Type: 100 \$10 Bracket 1
- Price Type: 101 \$10 Bracket 1
- Price Type: 102 \$10 Bracket 1

- Price Type: 200 \$10 Bracket 1 Target Price Type=100
- Price Type: 201 \$10 Bracket 1 Target Price Type=102
- Price Type: 202 \$10 Bracket 1 (not populated = all items)

Catalogue Item c

- Price Type: 100 \$10 Bracket 1
- Price Type: 101 \$10 Bracket 1
- Price Type: 102 \$10 Bracket 1
- Price Type: 200 \$10 Bracket 1 Target Price Type=100
- Price Type: 201 \$10 Bracket 1 Target Price Type=102
- Price Type: 202 \$10 Bracket 1 (not populated = all items)

6.4. Price Commentary

~~Need addition of code value PRICE_COMMENTARY to codelist special scenario (MP06) + add validation rules for Multiple Price:~~

A price commentary may be entered for all parent price types with an application sequence of 1. This rule explicitly excludes the following price types:

- ALLOWANCE
- CHARGE

The same price type must not be present in the parent price and commentary price. For example you cannot have a List Price in both the parent and commentary.

There must not be repeating prices of the same type in the commentary. For example, you cannot have two transactional prices within the commentary.

The use of commentary price should be managed at relationship level using the special scenario code of "Commentary Price".

The Catalogue Price Confirmation will determine when a price is acceptable.

There will be no validation rules against the special scenario code.

Note: The Price Commentary class has been removed from the Price Synchronisation message. In order to send price commentary information you should instead use the flexible extension documented in the following guide: [GDSN-Flexextend-ImpGuide](#).

~~Note: The best practice is to send a "commentary price" as a special scenario code when sending a commentary price.~~

~~If commentary exists but no special scenario code is sent, this can be handled by the data recipient with a confirmation status of Review.~~

6.7.6.5. Special Scenario Code

The Special Scenario Code List is a code list containing special price synchronisation scenarios that may be prevalent in certain target markets based on regional business practices or regulations. This list is used to trigger special processing by the data source and/or data recipient based on the needs of a scenario.

6.7.1.6.5.1. Resynchronisation of All Price Types for Pricing Done at the Lowest Level Consumer Unit

This scenario uses the NO_ACTION code for handling regional legal requirements for pricing to be done at the lowest level consumer unit.

In this scenario, the retailer can know the price of the product for ordering only once they have pricing for all the lowest level consumer units. When pricing changes, a retailer needs to understand when they can order the product. In this scenario, where pricing is done at the lowest level consumer unit, the retailer requires the synchronization of ALL pricing components of the ordering product (identified through the use of the Parent Catalogue Item attribute) when one or more component changes. Those that have new / changed pricing will use an action code of Add or Change. Those that have not changed need to be resynchronized with action code of NO_ACTION.

6.6. Target Market Information Provider

Target market information provider allows information providers to send the name and party identification of the local information provider of a price. To provide this information use targetMarketInformationProviderName and/or targetMarketInformationProviderPartyIdentification.

7. Appendices

Not Applicable

8. Summary of Changes

| Change | BSD Version | Associated CR Number |
|---|-------------|----------------------|
| <ul style="list-style-type: none"> Updated version number and date Updated all use case diagrams in document to be in line with Modelling Methodology for Major Release 3 Removed Rate From ComponentValueTypeCode Changed PriceSynchronisationConfirmationStatusCode to SynchronisationConfirmationStatusEnumeration Changed attribute name in ConditionTargetEntity to the priceSynchronisationCondition class to gpcCategoryCode. | 1.3.1 | n/a |
| <ul style="list-style-type: none"> Corrected typo in distributionMethodCode Added avpList association into ItemDepictionQualifier, PriceSynchronisationRelationship, PriceSynchronisationCondition. Corrected document status in cover and footers. Changed entity identification to top level for all classes. | 1.3.2 | n/a |
| <ul style="list-style-type: none"> PriceSynchronisationDocument: Changed priceDocumentTypeCode to priceDocumentType. PriceSynchronisationSegmentConfirmation: Changed priceSynchronisationConfirmationStatusCode to priceSynchronisationConfirmationStatus ItemPriceType: Moved Extension to last attribute (before AVPList) ItemPriceType: changed priceValueTypeCode to | 1.3.3 | |

| Change | BSD Version | Associated CR Number |
|--|-------------|----------------------|
| <ul style="list-style-type: none"> priceValueType ItemPriceType: Changed data type for AVPList PricePerformanceRequirementInformation: Changed performanceRequirementOptionCode to performanceRequirementOption PriceSynchronisationCondition: changed datatype for AVPList PriceSynchronisationRelationship: changed datatype for AVPList PriceSynchronisationRelationship: changed relationshipTradeChannelCode to relationshipTradeChannel | | |
| <ul style="list-style-type: none"> Removed line numbers Changed from "Draft" to "Issue" | 1.3.4 | |
| <ul style="list-style-type: none"> <u>Replaced the status of Accept to RECEIVED throughout the document since ACCEPT has been replaced by RECEIVED for 3.1.</u> <u>Changed all the references to the status codes in the document to all caps to match code value.</u> <u>Clarified section 6.4 eliminating the suggested need to send a Special Scenario Code.</u> <u>Changed Rule 9 in UC-11 regarding the use of Price Type Segment Action code of Delete.</u> <u>Added targetMarketInformationProviderName and targetMarketInformationProviderPartyIdentification to PriceSynchronisationDocument</u> <u>Added comments into model and section 6.4 stating that the Price Commentary class should not be used to send price commentary information.</u> <u>Deleted the association to the PriceCommentary Class from ItemPriceType. Deleted class.</u> Fixed link in section 6.4 | 1.3.5 | |