Data Recipient Best Practice

Delta Files vs. Block Updates

Prepared by Stephen Long

June 2025 Version 1.0

Disclaimer:

Please note: The information provided in this document is intended for guidance purposes only. Those involved in the creation, collection, management or distribution of product metadata are strongly advised to seek guidance on compliance with the business policies of their respective organisations.

Index

1. Introduction

An ONIX product record provides an accurate representation of that product at one point in time (and an ONIX message with multiple product records can do this for all the active products in a publisher's inventory). However, during the lifecycle of the product, the attributes of that product will change. The purpose of an ONIX update is to reflect the changing attributes in a timely and accurate way.

In this document, we describe the key differences between ONIX 'full files', 'delta files' and 'block updates'. All offer industry-recognised ways of updating product records. Those product records will have been originally provided as part of a full file (also referred to as a 'full feed' or a 'full load'), or as a record in a previous delta file.

Typically, an ONIX data feed begins with a 'full file' – that is, a complete product record for every product. Subsequent changes could be communicated in another full file – but since most product's attributes do not change on a daily or even weekly basis, the provision of 'delta files' is more common and much more efficient. Delta files contain updated product records for each product whose attributes have changed, as well as initial product records for any new products, plus potentially, a very small number of product records to be 'deleted', but they omit product records for any products where the metadata is unchanged.

Note that the structure of a full or a delta file is the same – it consists of a number of complete product records. For each record, a recipient should either update the existing data about that product in their database or add a new product to their database. The only difference is that a delta file omits product records for products where the recipient already holds up-to-date information.

However, the provision of an initial full file is not a one-off exercise. It can be useful to provide a new full file, perhaps annually or twice a year, to guard against any build-up of errors, 'reset' the active inventory and ensure that it is an accurate representation of the publisher's catalogue. Indeed, it is important to regularly confirm the current active inventory to avoid assumptions being made about a product's status.

'Block updates' are a refinement of a product record. In a delta file, product records are normally complete – they contain all the available metadata for that product, even if only one part of that metadata needs to be updated. These product records will include many or all of blocks 1–8 of an ONIX <Product>. In contrast, a block update record omits any of these blocks where the metadata remains unchanged. A single change by the data sender – for example, a change in Publishing status or a correction to the spelling of a contributor name – implies sending a single updated Block (although multiple blocks can be included if the changes span several blocks). A block update record also *always* includes 'block zero'. [NB a block update record makes no sense within a full file.]

The content and frequency of the full files, and of any delta files, should be agreed between the data sender and recipient. This ensures that they reflect the recipient's needs (for example, the recipient may only require a subset of records from the active inventory, limited to a specific product form). Similarly, whether the parties can cope with block updates rather than full product records needs to be agreed in advance.

2. Key differences between full product records in an ONIX delta file and block updates in a delta file.

a. A data sender must ensure that each Product record is assigned a unique and permanent Record reference. Ideally, this should not be the ISBN, as this can be problematic for recipients who may receive data about a single ISBN from more than one data source (e.g., from the publisher and the distributor). A suitable Record reference might be a UUID (a Universally Unique IDentifier), or a combination of sender's domain name and a row number from the data sender's database. Record references are used by recipients to recognise products that are new to the data feed, and Product records that represent updates of the metadata for an existing product. UUIDs can be obtained easily from www.uuidgenerator.net

b. An ONIX 'full file' contains a Product record for each of the sender's products (or at least each product within the agreed scope of the data exchange). An ONIX 'delta file' presupposes that the data recipient has previously received a full file. The delta contains only 'replacement records' for those products where some aspect of the metadata has been updated since the full feed, plus new records and possibly a very small number of records for deletion.

c. Product records in delta updates include the whole of the ONIX Product record. This includes those parts of the metadata that have been updated and those parts that remain unchanged.

Product records that are 'block updates' do not include the whole of the Product record. They omit sections (or 'blocks') of the metadata where there are no changes. Block updates represent an efficient way for data senders to highlight only those data elements within a product record that have changed, since they reduce (though do not eliminate) the amount of redundant, unchanged metadata in each updated record. Similarly, they can be an efficient way for data recipients to update product records, particularly where there are systems constraints.

Note that Block update records do not contain only the individual fields that have changed. They contain the whole Block(s) of data (that contain the change(s)) for a product, even where most of the fields in a block remain unchanged. They omit Blocks where there are *no* updates.

d. The use of block updates assumes that both sender and recipient organisations are familiar with the concept, how they differ from typical delta files, and that the systems capabilities exist (or can be developed).

Note that the difference between a full file and a delta file is about the selection of records included in the file. The full file contains all Product records within the agreed scope of the feed. The delta contains the subset of Product records that contain updated metadata, and/ or new records and/ or records for deletion. In contrast, the difference between a full Product record (in a Delta file) and a Block update product record is that the Block update omits some blocks of data from the record whereas a full Product record includes all blocks of data (even when a particular Block does not contain any updated data).

Full or Delta is a feature of the ONIX file. Full record or Block update is a feature of a particular Product record.

e. Full and Delta files use <NotificationType> codes 01, 02 and 03 within each Product record to indicate which part of the lifecycle each product is in. Block updates in a Delta file have their own Notification Type code [04 Update (partial)] in ONIX Code List 1.

f. Deltas (whether they contain full Product records or Block update product records) require the correct file sequencing. This relies on one of the following informing the order of receipt and processing:

   i. File name
   ii. Date and/ or time of receipt (however transmission failure and re-transmission can make this unreliable)
   iii. Ideally, the embedded date and time of file creation (see <SentDateTime> in the ONIX file Header)

If any ONIX file is inadvertently processed out of sequence (i.e., file 2 is missed out of the sequence of files 1, 2 and 3… then the data recipient *must* process file 2 before re-processing file 3 and any later files).

<SentDateTime> allows the sequence of data files to be confirmed. Note that a *missing* file cannot be inferred from <SentDateTime> except in cases where there is a rigid file delivery schedule such as 'exactly one file per day'. <MessageNumber> in the ONIX Header can contain a sequence number (1, 2, 3…) to make missing files obvious.

g. Remember also that there is a slight difference between ONIX 3.0 and 3.1 using Block 6, <ProductSupply>:

   i. In ONIX 3.1 a single market (as represented by a single <ProductSupply> composite) can be 'Block updated' without the risk of deleting the data for other markets (in other <ProductSupply> composites). The proviso is that you must name the markets to do this, using the <MarketReference> tag in each of the <ProductSupply> composites. This is useful if you have several Block sixes, one for each market, and want to uniquely target one for update.

   ii. In contrast, in ONIX 3.0, markets cannot be individually named. Multiple <ProductSupply> composites either require updating or reconfirming that the data has not changed as a single group. If there are three markets (represented by three <ProductSupply> composites), and one contains an update, you must send *all three*. If you send only one, the assumption is that the missing data (i.e., the other two) are no longer required and should be deleted.

   No other Block works this way.

   iii. If Market tags are not implemented, ONIX 3.1 is backwards compatible, so the 3.0 convention for handling updates to the <ProductSupply> composite will work in 3.1, but not vice versa.

3. Delta Files vs. Block Updates

#	Block Update	Delta File
1	Individual blocks of data (blocks 1 – 8), can be updated individually. Block 0 is always required as it contains the Record reference and the product identifier(s). Then send any block that requires an update. Blocks that are unchanged can be omitted. You cannot send a *part of* a Block. You must send a complete Block, even if most of it remains unchanged. Within a block update record, you cannot 'delete' Blocks 1 or 4. There is a mechanism to delete all data in Block 2 (for example), by sending an 'empty' Block 2 using <CollateralDetail/>.	To change any element in a record, the full product record needs to be submitted. Any change, however small, however frequent, requires a whole new record, incorporating the change. In principle, the new data completely overwrites the existing record.
2	If you omit a Block, it implies that all the data in that Block must be retained unchanged.	If you omit a Block, it implies that all the data in that Block must be deleted.
3	Block 1, Product Description and Block 4, Publishing Detail, cannot be deleted by way of block updates (i.e., sending an empty block 1 or 4), because they have mandatory elements for record integrity and maintaining a product's audit trail.	Beware that Block 1 (Product Description), and Block 4 (Publishing Detail) contain mandatory elements. Deleting Block 1 or 4 by omitting them from a Delta would make the ONIX record invalid and destroy the integrity of the record.
4	Subject to the above caveat, you should omit specific data in a block and update that block to remove that specific data. If you send an empty block, you will remove *all* the data in that block.	Removing data from a delta update will delete and remove that data from the new, updated record.

#	**Block Update**	**Delta File**
5	To delete a whole product record, simply include block 0 with a notification type of 05 for deletion. However, consider the implications of deletion and whether updating the publishing status could resolve the issue. In practice, almost all 'deletions' should be handled as changes of Publishing status. BIC has more information on Product Record Deletions on the BIC website.	To delete a record via a Delta, send only Block 0 with code 05 deletion request. There is no need to include blocks 1 – 8 since the whole point is that you want to delete the record. See the example in EDItEUR's Best Practice Guide by searching for **'deleting a product record that was issued in error'**. However, consider the implications of deletion and whether updating the publishing status could resolve the issue. In practice, almost all 'deletions' should be handled as changes of Publishing status. BIC has more information on Product Record Deletions on the BIC website.
6	A block is the smallest unit of data that can be updated.	A full product record is the smallest unit of data that can be updated.

[continued overleaf]

4. More About ‘Full Files’

Note these definitions of full files in the context of overall metadata management:

Distributors

A full feed should include all active titles and everything that has become inactive (because it is no longer available, or you no longer distribute it, or it has been cancelled) in the past 12 months. It should not include anything that has been inactive for more than 12 months (unless a different parameter has been agreed between data sender and data recipient).

Note the definitions for ‘active’ and ‘inactive’ below. These are taken from the current version of EDItEUR’s ONIX Code List 64, Publishing Status.

> ***Active:*** *The product was published and is still active in the sense that the publisher will accept orders for it, though it may or may not be immediately available.*
>
> ***Inactive:*** *The product was active but is now permanently or indefinitely inactive in the sense that the publisher will not accept orders for it, though stock may still be available elsewhere in the supply chain.*

All forthcoming titles that have been previously supplied should also be included. Unless by prior arrangement, the full feed should be delivered via your normal electronic file format.

Publishers

A full feed from a publisher should contain all active titles. Anything that has become inactive in the last 12 months should be included (unless a different parameter has been agreed between data sender and data recipient). All forthcoming titles that have been previously supplied (with territory pricing or market rights changes as required) should be included. Publishers should ensure that they provide product information that is as complete as possible for all areas of the record, including descriptive content. Unless by prior arrangement, the full feed should be delivered via your normal electronic file format. Full feeds should ideally be sent by prior arrangement between the sender and recipient.

Regular updates of product records should be provided even if nothing has changed in terms of price and availability. It is important to regularly confirm current price and availability status to data aggregators and all other trading partners to avoid assumptions being made about products’ status.

Do not create or use proprietary codes in house. These risk being included in files to trading partners, causing confusion. Similarly, do not misappropriate or misuse existing, standards-based codes.

The requirement in relation to out of print (OP) titles should be to report products as OP up to one year after the actual OP or permanently withdrawn from sale (PWfS) date, rather than to simply stop sending the data. Removing the audit trail immediately would cause confusion, generate unnecessary queries and create ambiguity. Remember that ‘out of print’ could mean ‘unavailable’ or that ‘*existing* stock is still available in the supply chain, but no *new* stock is available’.

The sending of files needs to be continuous, consistent and in the correct sequence (mirrored by receiving and processing routines).

5. Resources

EDItEUR has an extensive range of related resources available on its website:

ONIX release 3.0 and 3.1 Downloads -https://www.editeur.org/93/Release-3.0-and-3.1-Downloads/#Specifications

Application Note on Block Updates in ONIX 3.0 and 3.1, including examples - https://www.editeur.org/files/ONIX%203/APPNOTE%20Block%20updates%20in%20ONIX.pdf