EDItEUR

ONIX for Books release 3.0.8

This document summarises the updates incorporated into ONIX 3.0 revision 8. These changes were ratified in outline by the ONIX International Steering Committee at its meeting held by videoconference at the time of the Frankfurt Book Fair (October 2020), and details were confirmed by the subsequent Steering Committee conference call at the end of April 2021. The changes comprise a major new ONIX data block to carry information about production of products in the print on demand, e-book and digital audio supply chains.

For clarity, these updates are all optional and fully backwards compatible, and there have been no new deprecations. All ONIX messages that were valid under previous iterations of ONIX 3.0 remain valid according to the latest 3.0.8 Specification and the latest XML validation tools. Full use of this release requires Codelists Issue 53a [1] or later, and the necessary codelist files are available from the EDItEUR website (https://www.editeur.org/14/code-lists) and via the online multilingual ONIX codelist browser (https://ns.editeur.org/onix).

1. Production detail

The primary innovation in ONIX 3.0.8 is the addition of a new Block 8, which in message order occurs between Blocks 5 and 6.

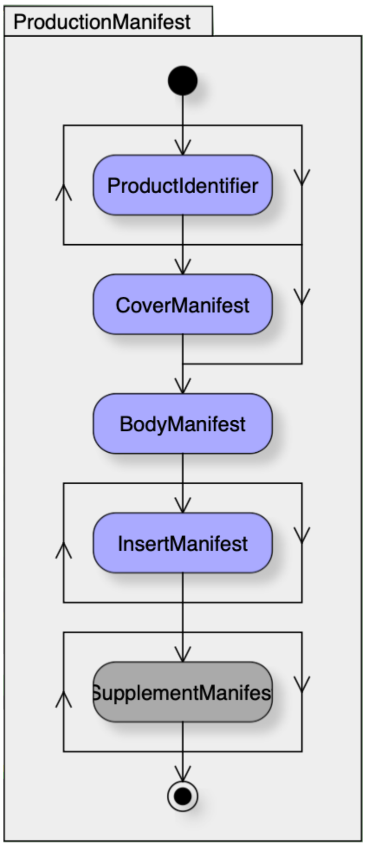
Block 8 – the <ProductionDetail> block – is intended to carry detail relating to intermediary services within the supply chain, for example print on demand (POD), e-book conversion services or distribution of digital audio. In these cases, there are publishers and service providers who make use of ONIX to exchange bibliographic, marketing and commercial metadata, and they can use Block 8 to eliminate the use of additional 'sidecar' files that might otherwise need to be distributed by the publisher alongside the ONIX and the various content files.

It is not expected that all, or even a large proportion, of ONIX messages will contain Block 8 – most will omit it, and data senders who choose to use Block 8 may in fact omit it from ONIX sent to 'conventional' data recipients. Only a few ONIX messages – predominantly those targeted at POD, e-book and audio service providers – might be expected to include it. However, all ONIX recipients should be prepared to receive or pass through Block 8: even if they have no interest in the data it can contain, the Block as a whole can be ignored.

The Block consists of a repeatable <ProductionManifest> composite, each repeat of which encapsulates specification and file manifest details for the product or for a single product part. Products with multiple parts should include multiple <ProdutionManifest> composites [2].

[1] Issue 53a is an interim release where most codelists are unchanged from Issue 53, but a handful of new codelists have been added to support new data fields in ONIX 3.0.8. Following release of Issue 54, 53a will simply be known as 'Issue 53'.

[2] The link between each product part and each production manifest is maintained by a <ProductIdentifier> within the manifest that must match one within <ProductPart>. In many cases, this will be some proprietary identifier for the part, or a simple UUID.

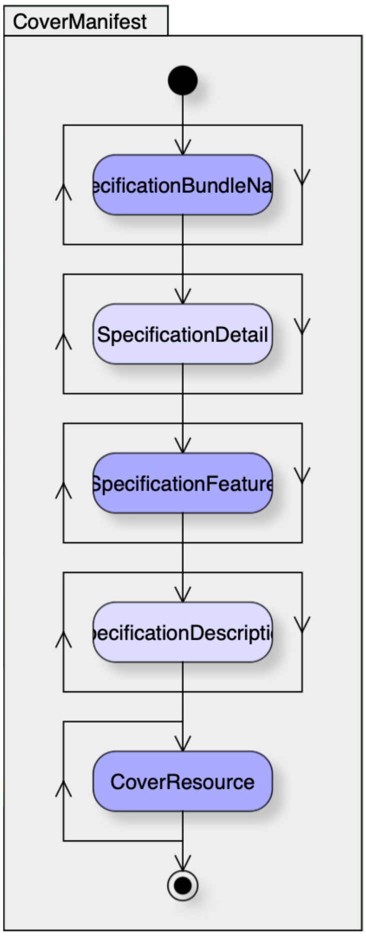


Within a Production manifest, there must be some combination of four types of manifest – cover, body, insert and supplement (see diagram left). Each contains details of the relevant section of the product (or product part), and of course, depending on the overall form of the product, not all manifest types are relevant – digital audio products will not include a cover manifest, for example, and most POD books will not include an insert (a 'plate section').

Each of the first three types of manifest – cover, body and insert – has roughly the same structure and contains three types of data:

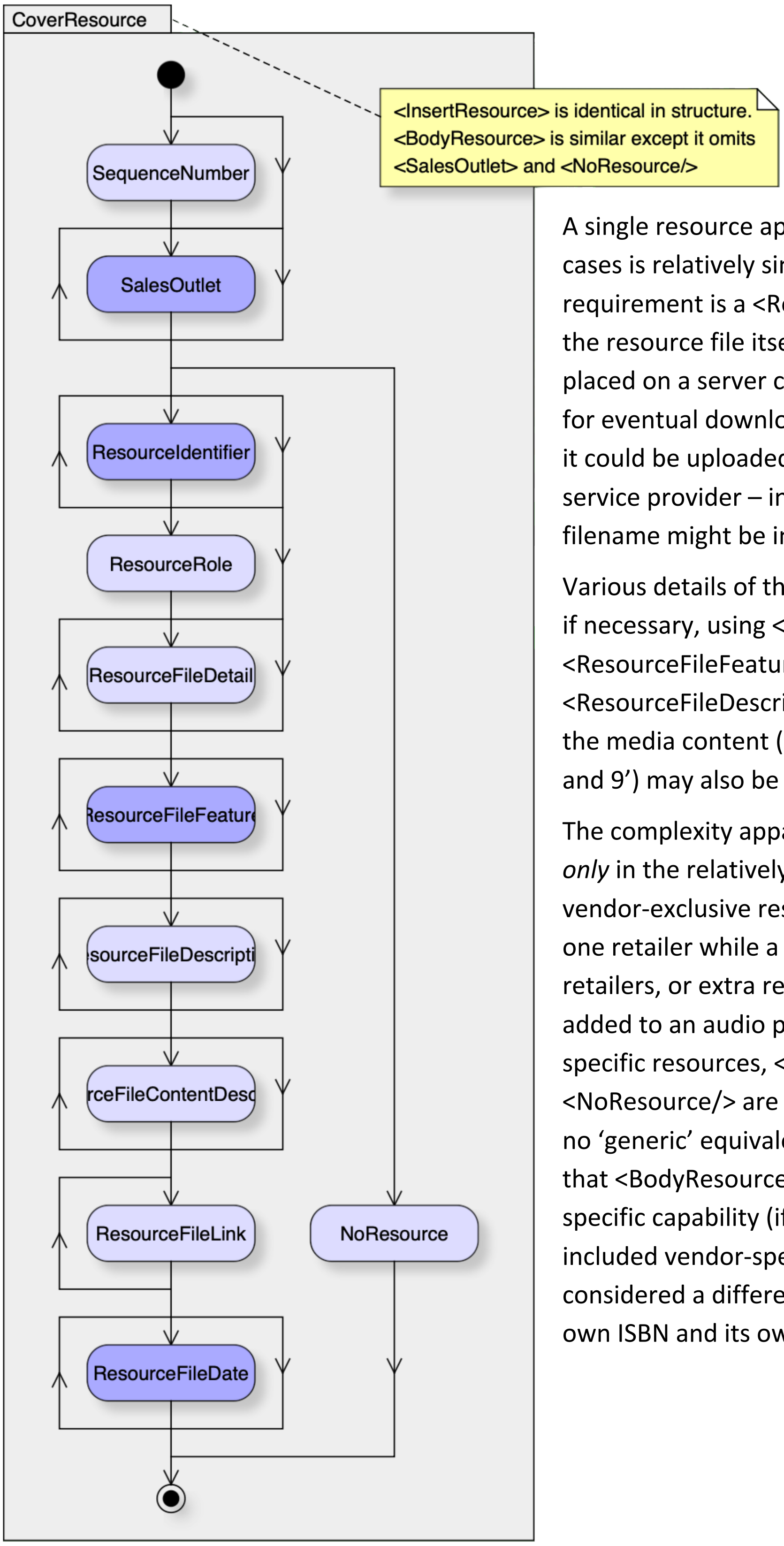
- any proprietary specification 'bundle'. Often a POD manufacturer supplies books to only a small number of specifications – combinations of a specific size, paper type binding type and so on, a predefined bundle of specs. Or;
- in the absence of a particular bundled spec, any detailed specifications necessary for that section of the final product can be supplied using <SpecificationDetail>, <Specification-Feature> and possibly <SpecificationDescription>;
- a manifest of the files necessary to create that section of the final product (for example WAV files to be turned into the digital audio, or PDFs for print on demand.

The structure of a single cover manifest is shown in the diagram lower left.



The Supplement manifest is different: it's intended for items that are supplementary to the product but not listed as Product parts – for example a small PDF booklet that might be downloaded alongside a digital audio product. So the structure of <SupplementManifest> is different – it consists mostly of Cover, Body and Insert manifests nested inside the Supplement manifest, together with <ProductForm> and <ProductFormDetail> for the supplement.

One of the primary requirements of each manifest is that it contains a list of resources – the files required to create a section of the final product. The structure of a single resource is shown in the diagram overleaf.



<InsertResource> is identical in structure.
<BodyResource> is similar except it omits
<SalesOutlet> and <NoResource/>

A single resource appears complex but in most cases is relatively simple. The only mandatory requirement is a <ResourceFileLink>, the URL of the resource file itself. This file would normally be placed on a server controlled by the ONIX sender, for eventual download by the service provider, or it could be uploaded to a server controlled by the service provider – in the latter case, a simple filename might be included instead of a full URL.

Various details of the resource file can be included if necessary, using <ResourceFileDetail>, <ResourceFileFeature> and possibly <ResourceFileDescription>. A brief description of the media content ('this PDF contains chapters 8 and 9') may also be included.

The complexity apparent in the diagram is used *only* in the relatively rare case of vendor-specific or vendor-exclusive resources – a unique cover for one retailer while a generic cover is used for other retailers, or extra retailer-exclusive end matter added to an audio product. For such vendor-specific resources, <SalesOutlet> and <NoResource/> are used – the latter when there is no 'generic' equivalent for other retailers. Note that <BodyResource> does not have this vendor-specific capability (if the body of a product included vendor-specific material, it should be considered a different product altogether, with its own ISBN and its own ONIX Product record).

Using a number of repeats of <CoverResource>, <BodyResource> or <InsertResource>, all the files necessary to print, convert or assemble the complete product can be listed.

2. Other minor additions

2.1 The <TelephoneNumber> field has been added to <ProductContact>, <SupplyContact>, <Sender> and <Addressee> composites. This allows a cleaner listing of contact's telephone numbers where previously (by convention) they were carried alongside a name in the contact name field. In these new contexts, the <TelephoneNumber> is not repeatable. Implementers should note that full international dialling details are recommended, *ie* +44 20 8843 8607 rather than (0)20 8843 8607 (the +44 international dialling code indicates the UK, whereas the (0) is used only for calls within the UK).

2.2 An <AudienceHeadingText> data element has been added within the <Audience> composite, for use with proprietary audience coding schemes, or to include heading text alongside codes taken from established schemes. Note that for proprietary audience schemes, <AudienceCodeType> should be 02, whereas in many other ONIX codelists, code 01 indicates a proprietary scheme. For proprietary schemes, <AudienceCodeTypeName> is mandatory.

2.3 A repeatable <SupportingResources> composite modelled on that from Block 2 has been added to Block 7 – <PromotionDetail> – to provide links to marketing resources such as posters, flyers, reading guides *etc* in support of specific promotional events or individual event occurrences.

3. New documentation

The updated *Specification* for ONIX 3.0 revision 8 is now available from the EDItEUR website (https://www.editeur.org/93/Release-3.0-Downloads/), together with the necessary XML schemas (including an updated 'strict' schema) and Issue 53a of the ONIX Codelists. Updates to the ONIX 3.0 *Implementation and Best Practice Guide* will follow.

Graham Bell, Chris Saynor
EDItEUR
5th July 2021