



ONIX Mapping and Control Records

Prepared For

Iptor IP1

Revision Summary

Version	Notes	By	Date
1.0	Original	Michael Coon	12/12/2022
1.1	Review of every segment	Michael Coon	13/02/2023
1.2	Redo IC-IN IC-SUB, record id, character errors	Michael Coon	21/03/2023
1.3	OI1ILX control records, Email reports	Michael Coon	30/05/2023

Table of Contents

Revision Summary	2
Introduction.....	4
High level Process.	4
Mapping and Control Records	6
Product Identifier Type	6
Product Form Type.....	7
ONIX Segment Groups	10
Record Id.....	13
Publisher/Imprint IL1 Class IL2 Div IL3 Grp.....	14
Collection/Series	16
Nonstandard character mapping.....	18

Introduction

The ONIX feed from the presses is mapped to the IP1 product files by Iptor Exchange. The configuration for this mapping is done from the control files application ONIXAIDA.

The ONIX feed contains codes and values. The codes are defined in code lists. These code lists have been copied into the control records. Within the control record a code can be assigned one of the following statuses:

- Blank = the code is expected, and the associated value should be accepted
- * = the code is expected but a decision has been made that it can be ignored
- ? = the code is not expected and should be treated as an exception. If the code is received a decision needs to be made and the status changed to either *blank* or *. If changed to *blank* additional changes may need to be made to process the value correctly. Generally, for all code lists the status is set to '?' when created.

The control records come in three basic formats.

1. **Code list** = this is a direct copy from the code list. The record will have a name of ONIXxxx where xxx is the number of the code list.
2. **Text** = used to hold the names of the publishers, presses and collections/series. These can span multiple lines. The names are ONIXPUB ONIXIMP and ONIXSER. ONIXABV maps long common strings in publisher or imprint names to an abbreviation.
3. **Field name** = these records are used when the same code list is used to set multiple fields, or a field is set from a combination of code lists. E.g. OI1ILX sets the field IL1/IL2/IL3 I1DIV based on combination of the values from ONIXPUB and ONIXIMP.

High level Process.

Each press sends files of ONIX data to its own specific address, to be processed by AIDA. AIDA assigns the data a batch number which is the value of the 'sentdatetime' segment in the feed with a three character identifier for the press. AIDA identifies segments and combinations of segments which are of interest and generates a record for each field to be updated. If the data cannot be created correctly or is to be ignored then status codes are set on the record. These records are bundled into a Cfile.

When the Cfile is received it is processed and IP1 is updated. When processing is finished an email will be sent to specified email addresses. Multiple copies of the email can be sent with different conditions for each copy.

The subject of the email will begin with either

- ONIX COMPLETED
- ONIX ACTION REQUIRED

If action is required then check the attachments in the email. These attachments are csv files and should open into excel.

Attachment	Purpose
pubimprep.csv	When the incoming imprint or publisher is not on file. The records are created automatically, this report is an audit trail and suggest what else needs to be done.
seriesrep.csv	When an incoming collection is not on file. The records are created automatically, this report is an audit trail and suggests what need to be checked.
otherrep.csv	List other mapping errors. The report contains suggestions as to what should be done.
validationrep.csv	List validation errors. These are generated at the time of updating the product and other file. The report contains suggestions of what needs to be done. There may be an overlap with the otherrep.csv where an error in the mapping may cause a validation error.
charrep.csv	Character errors. This lists nonstandard characters, usually those with accents which have not been handled properly. It is expected that Iptor will correct these.

When all the issues are fixed, which will generally require changes to control records, the batch will need to be reprocessed. Reprocessing sends the ONIXAIDA control records to AIDA along with a request to reprocess the batch.

AIDA will reprocess the original ONX file using the updated data from the control records and send a new CFile to be processed. At the end of processing a new email is sent. The turnaround from request to email is variable and could be 5 to 15 minutes.

If there are still errors repeat the process or ask Iptor for help.

To reprocess take option 'Work with ONIX import batches ONI020' on menu

- Other Regions
 - o US Options
 - EDI – AIDA Onix
 - Onix Management

Right click on the batch number which was on the email and click reprocess.

Mapping and Control Records

Product Identifier Type

This is an extract from an ONIX feed, showing different product identifiers for a single product.

```

<ProductIdentifier>
  <ProductIDType>02</ProductIDType>
  <IDValue>0803249306</IDValue>
</ProductIdentifier>
<ProductIdentifier>
  <ProductIDType>03</ProductIDType>
  <IDValue>9780803249301</IDValue>
</ProductIdentifier>
<ProductIdentifier>
  <ProductIDType>15</ProductIDType>
  <IDValue>9780803249301</IDValue>
</ProductIdentifier>

```

The definition of the ProductIDType can be found on the code list. Online search 'ONIX ProductIDType' will return:

List 5: Product identifier type

Last updated in issue 14

Value	Description	Notes
01	Proprietary	For example, a publisher's or wholesaler's product number or SKU. Note that <IDTypeName> is required with proprietary identifiers
02	ISBN-10	International Standard Book Number, pre-2007 (10 digits, or 9 digits plus X, without spaces or hyphens) - now DEPRECATED in ONIX for Books, except where providing historical information for compatibility with legacy systems. It should only be used in relation to products published before 2007 - when ISBN-13 superseded it - and should never be used as the ONLY identifier (it should always be accompanied by the correct GTIN-13 / ISBN-13)
03	GTIN-13	GS1 Global Trade Item Number, formerly known as EAN article number (13 digits, without spaces or hyphens)
04	UPC	UPC product number (12 digits, without spaces or hyphens)
05	ISMN-10	International Standard Music Number, pre-2008 (M plus nine digits, without spaces or hyphens) - now DEPRECATED in ONIX for Books, except where providing historical information for compatibility with legacy systems. It should only be used in relation to products published before 2008 - when ISMN-13 superseded it - and should never be used as the ONLY identifier (it should always be accompanied by the correct GTIN-12 / ISMN-13)
06	DOI	Digital Object Identifier (variable length and character set beginning '10', and without https://doi.org/ or the older http://dx.doi.org/)
13	LCCN	Library of Congress Control Number in normalized form (up to 12 characters, alphanumeric)
14	GTIN-14	GS1 Global Trade Item Number (14 digits, without spaces or hyphens)
15	ISBN-13	International Standard Book Number, from 2007 (13 digits starting 978 or 9791-9799, without spaces or hyphens)

This code list is 'List 5'. All code lists have a numeric identifier. This code list has been copied into the control file as ONIX005 using the list number as an identifier:

ONIX Segment Groups

Table of ONIX segment groups and how they are mapped:

	Control Record	Id/ File	Target Field	
NotificationType	ONIX001	O01 DSIM00P	I1STS	NotificationType is a code, it is mapped to a different code. Field must be I1STS. Value validated against TMSDS/IM-I1STS.
ProductIdentifier ProductIdType IdValue	ONIX005	O01 DSIM00P	I1SUPI I1AI I1I	Convert ProductIdType to a field and put ProductIdValue in the field. Valid values for the field are I1I, I1AI and I1SUPI.
ProductForm ProductFormDetail	ONIX150 ONIX175 I1CAT I1CATE I1PKG I1PKGE I1STKF I1STKG	O14 O15 O16 DSIM00P	I1CAT/IM- LVL4 I1PKG I1STKF	Both segments are checked for validity against their codelist. For each target field the pair of elements are checked against the exception codelist ending in E and if no match is found it tries for a match against the generic codelist. Validated again TMSDS/IM-IL4, TMSDS/IM-PKG, TMSDS/IM-STKF.
CountryOfManufacture	ONIX091	O01 DSIM00P	I1ORGN	Converts code to value in TMSDS/IM-ORGN. If CountryOfManufacture not supplied I1ORGN is defaulted when the DSIM00P record is created.
CollectionType TitleType TitleElementLevel TitleText	ONIX148 ONIX015A ONIX149A ONIXCOL ONIXSER	O23 DSIC00P	I5ICT I5ICC	TitleType and TitleElementLevel are used outside collection with different mapping. The suffix A is the version for collection. ONIX148 selects the relevant CollectionType. ONIX015A sets I5ICT=IC-SER for the relevant values. ONIX149A set the target field to I5ICC for relevant values. TitleText is mapped to a code using ONIXCOL contains the full text of the series and id. ONIXSER connects the collection id from ONICCOL with the series ID in TMSDS/IC-SER. When a new series is found it makes multiple attempts to match removing commas, full stops and colons. If no match is found, all three records are automatically created. Text is converted to upper case, but must be reviewed before the batch is reprocessed to complete the update.
TitleType TitleElementLevel TitleText	ONIX015 ONIX149	O18 DSIM00P	I1IDSC	TitleType/ONIX015 is for selection. TitleElementLevel/ONIX149 is for selection. TitleText goes to I1IDSC.

				UPPERCASE	
Subtitle	ONIX015 ONIX149	O18 DSIM00P	I1EDC	TitleType/ONIX015 is for selection. TitleElementLevel/ONIX149 is for selection. TitleText goes to I1EDC UPPERCASE.	
TitleWithoutPrefix	ONIX015 ONIX149	O18 14DSIM00P	I1IDSC	TitleType/ONIX015 is for selection. TitleElementLevel/ONIX149 is for selection. TitleWithoutPrefix goes to I1IDSC UPPERCASE Ignored if I1IDSC already set from TitleText.	
Contributor SequenceNumber				This is an id, used to link the following contributor fields which go to multiple files.	
ContributorRole PersonName	ONIX017	O04 DSIME00P	BICT	BISEQ= contributor sequence BICT =ContributorRole mapped from ONIX017 BIFNM = PersonName	
NameBeforeKey		O19 CRCMA00p 20 CRCMA00P	D1MDNM D1GVNM	Characters up to 1 st blank go to D1GNVNM given name Characters after that go to D1GVNM middle name.	
KeyNames		O05 CRCMA00P	D1FNM		
SuffixToKey		O05 CRCMA00P	D1NMSFX		
TitleBeforeNames		O05 CRCMA00P	D1SAL		
LanguageRole LanguageCode	ONIX022	O01 DSIM00P	I1LANG	LanguageRole/ONIX022 selection UPPERCASE.	
ExtentType ExtentValue ExtentUnit	ONIX023 ONIX024	O08 DSIM00P O09 DSIM00P	I1EXTV I1EXTM	ExtentType/ONIX023. Used as a selection for the other two. ExtentValue direct to I1EXTV ExtentUnit/ONIX024 selection and mapped to I1EXTM.	
SubjectSchemeIdentifier SubjectCode SubjectSchemeName	ONIX027 ONIX027B ONIXSST ONIXINT ONIXSUB	O13 DSIC00P O21 DSIC00P O22 DSIC00P O26 DSCIC00P	I5ICT I5ICC	Each subject code is written to 3 recid and if it satisfies the condition of the id it will update. Recid O13 Non Bisac SSI ONIX027 allows value 24 SSN ONIXSST sets up IC-INT if it contains substring Interest Code or IC-SUBM if it contains Minor Disc Recid O21 BISAC IC-INT SSI ONIX027B allows code 10 only Map SubjCode through ONIXINT Recid O22 BISAC IC-SUBM SSI ONIX027B allows code 10 only Map SubjCode through ONIXSUB Recid O26 IC-SUBJ SSI ONIX027B allows code 10 only 1 st 8 characters of subject code	

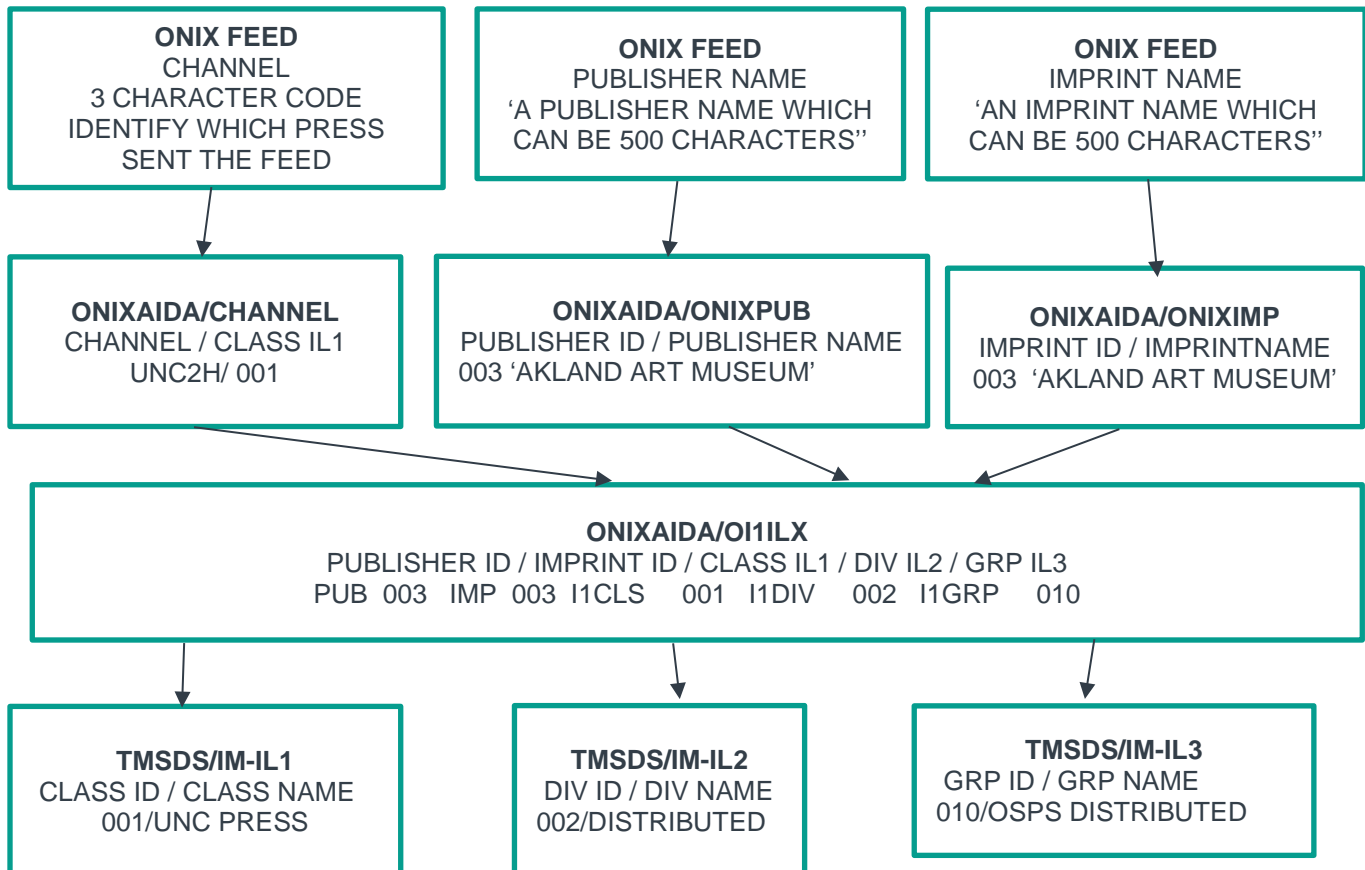
EditionNumber		O01 DSIM00P	I1RLSN	
PublisherName ImprintName	ONIXIMP ONIXPUB ONIXABV OIILX PPRESS OI1ILX OIILX OIC-ROY	O17 DSIM00P O11 I1DIV DISM00P O12 I1GRP DSIM00P O13 DSIC00P	I1CLS I1DIV I1GRP I5ICC	ONIXPUB and ONIXIMP both validate the publisher / imprint expected and use ONIXABV to abbreviate blocks of text, so the name is less than 70 characters. The following tables use the publisher id from ONIXPUB or imprint id from ONIXIMP to map these to their relevant IP1 field values. OI1PRESS maps the press from which the file was received from to item level 1. OI1ILX contains the expected press for the publisher OI1ILX maps publisher imprint to DIV (IL2) OI1ILX maps publisher imprint to GRP (IL3) OIC-ROY maps publisher to I5ICC this allows for a default value.
CountryofPublication	ONIX091	O01 DSIM00P	I1ORGN	I1ORGN maps the CountryofPublication to the same value
PublishingStatus	ONIX064	O01 DSIM00P	I1STKR	Validated against TMSDS/IM-STKR but this is empty.
SalesRightsType CountriesIncluded	ONIX046	O03 DSIC00P	I5ICT I5ICC	SalesRightType/ONIX046 determines I5ICT = IC-RGHTI included, IC-RGHTI excluded. The countries included is a list and copied unchanged.
ProductRelationCode ProductIdType IDValue	ONIX051 ONIX005A	O02 DSIMB00P	BJLNKI	ProductRelationCode and ProductType are for selection. IDValue is put in BJLNKI.
ReturnsCodeType ReturnsCode	ONIX053 ONIX066	O02 DSIMB00P	BJFSIF	ReturnsCodeType/ONIX053 is for selection. ReturnsCode/ONIX066 maps to BJFSIF.
SupplyDateRole Date	ONIX166	O02 DSIMB00P	BJRLSD	SupplyDateRole/ONIX166 sets the field. Date is the value
PriceType DiscountCodeType DiscountCode	ONIX058 ONIX100	O03 DSIC00P	I5ICT I5ICC	PriceType/ONIX058 is for selection. DiscountCodeType sets the I5ICT to IM-IDIS. The discount code is converted to upper case by the mapping and put in I5ICC.
PriceType PriceDateRole CurrencyCode	ONIX058 ONIX173 ONIX096	O06 DSPRC	I9EFFD I9EXPD I9RRP	PriceType/ONIX058 determines the field RRP PriceDateRole/ONIX174 determines EFFD EXPD CurrencyCode/ONIX096 is for selection

Record Id

Multiple segments cause updates to the same file. Some segments can trigger multiple updates to the same file. The record id can help to identify what caused the update when looking at multiple updates for the same file. The record id is a number 01 to 27 prefixed by the letter O.

O01	DSIM00P Item master general
O02	DSIMB00P Item by customer group
O03	DSIC00P Item classifications
O04	DSIME00P Item Contributor
O05	CRCMA00P Contact master file
O06	DSPRC00P pricing
O07	ONIMA00P Onix text - not used
O08	DSIM00P Extent type ONIX023 unit ONIX024 value
O09	DSIM00P Extent Type ONIX023 unit ONIX024 unit
O10	DSIM00P I1GRP Imprint Publisher OI1GRPE OI1GRP
O11	DSIM00P I1DIV Publisher Imprint OI1DIVE OI1DIV
O12	DSIC00PIC-ROY Publisher Imprint OIC-ROY
O13	DSIC00P IC-SUBJ/INT SSI = 24 ONIX027, SSN ONIXSST, SUBJECT
O14	DSIM00P I1CAT PRODFORM ONIX150, PRODFORMDET ONIX175,OI1CATE OI1CAT
O15	DSIM00P I1PKG PRODFORM ONIX150, PRODFORMDET ONIX175,OI1PKGE OI1PKG
O16	DSIM00P I1STKF PRODFORM ONIX150, PRODFORMDET ONIX175,OI1STKFE OI1STKF
O17	DSIM00P I1CLS Publisher Imprint OI1CLSE OI1CLS
O18	DSIM00P I1IDSC I1EDSC ONIX149,TitleWithoutPrefix,text,subtitle
O19	CRCMA00P D1MDNM Names before key ,any text after first blank
O20	CRCMA00P D1GVNM Names before key ,any text before first blank
O21	DSIC00P IC-INT BISAC SSI ONIX027B subject code
O22	DSIC00P IC-SUBJ BISAC SSI ONIX027B subject code
O23	DSCIC00P IC-SER Collection
O24	DSIMB00P BJSTKSTS
O25	DSIM00P I1STKR
O26	DSIC00P IC-SUBJ
O27	DSIM00P I1FRCHG

Publisher/Imprint IL1 Class IL2 Div IL3 Grp



Each press sends its onix feed via its own channel. The channel is mapped to a IM-IL1 class using ONIXAIDA/CHANNEL.

The text for each publisher is stored in ONIXAIDA/ONIXPUB along with an identifying code.

The text for each imprint is stored in ONIXAIDA/ONIXIMP along with an identifying code.

ONIXAIDA/OI1ILX maps from a pair of publisher and imprint identifying codes to:

- IL1 Class; this is linked to the channel which gives a class. If the class from the channel is not what is expected from ONIXAIDA/OI1ILX for the publisher/imprint it is an error
- IL2 Div; maps to TMSDS/IM-IL2.
- IL3 Group; maps to TMSDS/IM-IL3. Multiple imprints can map to the same group

If a new publisher and/or imprint are known in advance

- Create the ONIXAIDA/ONIXPUB and/or ONIXAIDA/ONIXIMP. Use the next available sequential number as an id.
- If it is a new imprint add a new TMSDS/IM-IL3 if necessary.
- Create ONIXAIDA/OI1ILX link publisher/imprint id with IL1 class, IL2 DIV and IL3 Group

If a new publisher and/or imprint are received without notice

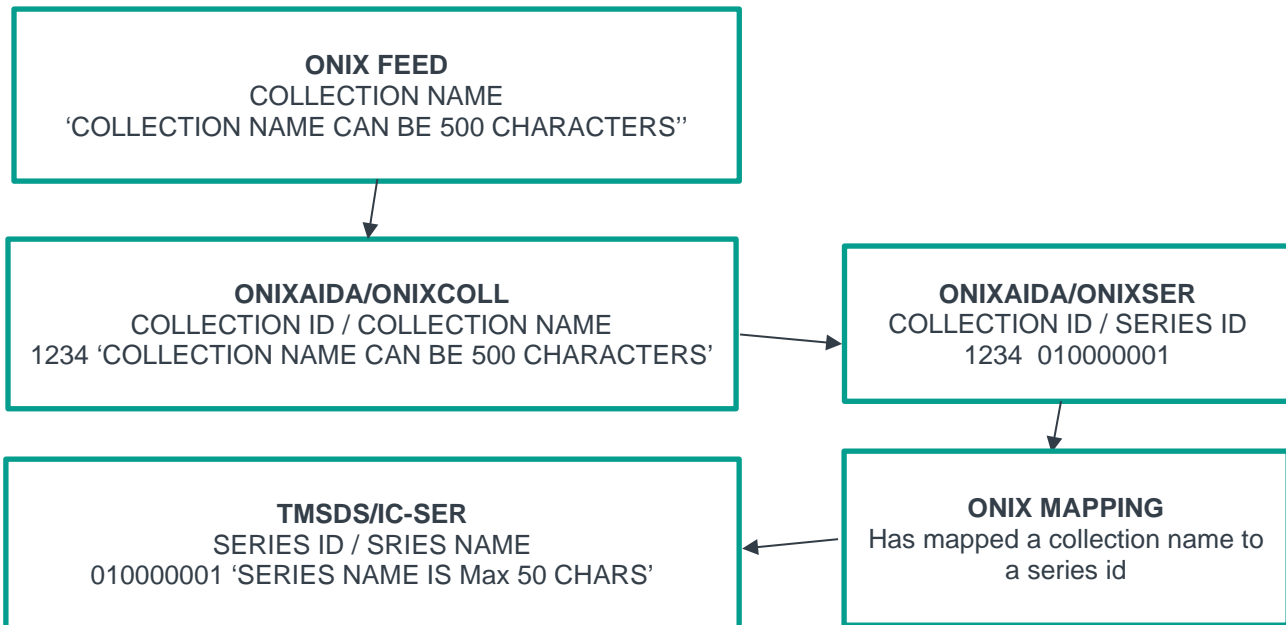
The mapping will record an error and it will be emailed on the Onix report in the pubimprep.csv file.

ONIXAIDA/ONIXPUB and/or ONIXAIDA/ONIXIMP will be created as necessary.

ONIXAIAD/OI1ILX will be created for the publisher/imprint combination. It will have ??? as values for IL1 class, IL2 DIV and IL3 Group. It may be necessary to create a new TMSDS/IM-IL3 for the imprint. The ??? values can be corrected and the batch must be reprocessed.

The program ONX153 Publisher/Imprint relationships will email a csv file with the relationships.

Collection/Series



The text for each collection is stored in ONIXAIDA/ONIXCOL along with an identifier.

Each collection must be matched with a record in TMSDS/IC-SER which has much shorter text and its own identifier. The text in TMSDS/IC-SER is what the user sees. A collection can appear in the feed with multiple different variations and abbreviations but they should all be able to map to the same TMSDS/IC-SER.

ONIXAIDA/ONIXSER links the ONIXAIDA/ONIXCOL identifier with the TMSDS/IC-SER identifier.

The Onix feed sends a collection name, which is compared with the collections in ONIXAIDA/ONIXCOL to find a match. The comparison is done in upper case and punctuation is ignored. If a match is found the id on the ONIXAIDA/ONIXCOL is used to access ONIXAIDA/ONIXSER to get the id of the TMSDS/IC-SER record.

No matching ONIXAIAD/ONIXCOL found for the incoming collection name

The mapping will record an error and it will be emailed on the onix report in the seriesrep.csv file.

The ONIXAIDA/ONIXCOL, ONIXAIDA/ONIXSER and TMSDS/IC-SER records will be created.

The id for the ONIXAIDA/ONIXCOL will be the next sequential number.

The id for the TMSDS/IC-SER will be three three character CLS/IL3 followed by a 5 digit sequential number.

It is necessary to check the generated TMSDS/IC-SER text:

- Has been truncated. In which case an abbreviated version of the full text in ONIXAIDA/ONIXCOLL can be keyed into TMSDS/IC-SER.
- Does it contain any escape sequences representing nonstandard characters.eg ' represents an apostrophe. Other escape sequences could represent a character with an accent. These escape sequence can be replaced with a suitable character e.g the character without an accent. These text is stored in a non unicode field so unlike titles and contributors it cannot hold characters with accents
- ONIXAIDA/ONIXCOLL already contained a record for this collection with slightly different text. This can be checked by running ONIX154 Collection/Series relationships. This will email a csv file containing all the collection/series which can be sorted on collection text to see if the new collection text is equivalent to an existing one. If it is either
 - o advise the press to fix the description and delete the newly created ONIXAIDA/ONIXCOL, ONIXAIDA/ONIXSER and TMSDS/IC-SER
 - o or change the new ONIXAIDA/ONISSER to point at the original TMSDS/IC-SER and delete the new TMSDS/IC-SER.

Once the records are ok, take the reprocess option for the batch. If the press has been asked to fix the collection do not reprocess but close the batch.

Nonstandard character mapping

It is expected that Iptor will handle nonstandard character issues.

There are 4 different character encoding standards involved in the mapping

The Onix feeds are in UTF8. This is an extension of ASCII which uses single byte characters and can only handle about 100 printable characters. In UTF8 standard characters are sent as single byte ascii, other characters such as those with accents are sent as two or three bytes. The IBM computer doing the mapping is using an IBM standard EBCDIC which can handle about 200 characters. It can handle character sets for multiple languages but only one at a time. The characters coming in for the contributor names do not fit into any single EBCDIC character set. The IBM IP1 matching can handle Unicode in which all characters are double byte.

For example

- ASCII/UTF8 The hex value 41 represents A
- UTF8 hex value C5A0 represents Š = S with caron
- Some of the presses send the nonstandard characters as escape sequences
- Š Means use the UNICODE character with decimal value 352 which is an Š.

The objective of the mapping is to convert nonstandard characters to escape sequences. This means that after the mapping the data will be consistent regardless of how the presses send the data

When the feed is copied into the mapping ascii codes are converted to EBCDIC codes, the hex value of the multi byte characters is altered but remains a recognisable multi byte character.

Identify mapping problem

When the feed for a title or contributor name contains a nonstandard character the mapping attempts to convert it to an escape sequence. If it cannot find a mapping in ONIXCHR it will return an error.

For the ISBN 9780806155142

l=93,a=81,-=60,Å=67, =41,a=81, =40, =40, =40, =40, The Life and Legacy of Z

These are the characters and their EBCDIC hex values around the problem character and the first 25 characters of the string with the problem.

Find the original in the feed

From the client, view the feed and search for the text, or search for the item and manually look for the text with a nonstandard character.

```
<TitleDetail>
```

```
  <TitleType>01</TitleType>
```

```
  <TitleElement>
```

```
    <TitleElementLevel>01</TitleElementLevel>
```

```
    <PartNumber>Volume 67</PartNumber>
```

```
    <TitleText>Red Bird, Red Power</TitleText>
```

```
    <Subtitle>The Life and Legacy of Zitkala-Ša</Subtitle>
```

```
</TitleElement>
</TitleDetail>
```

With this information we can see the problem character is between ‘-’ and ‘a’
 l=93,a=81,-=60,Å=67,=41,a=81,=40,=40,=40,=40,
 so the hex code for the character within the mapping is 6741

Identify the Unicode escape sequence

Search for Unicode Š

Compare website will show

Unicode Character “Š” (U+0160)

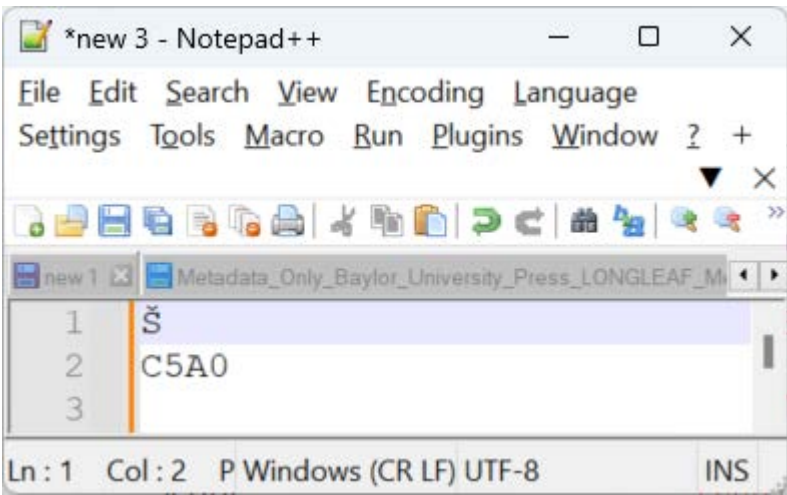


Name:	Latin Capital Letter S with Caron ^[1]
Unicode Version:	1.1 (June 1993) ^[2]
Block:	Latin Extended-A, U+0100 - U+017F ^[3]
Plane:	Basic Multilingual Plane, U+0000 - U+FFFF ^[3]
Script:	Latin (Latn) ^[4]
Category:	Uppercase Letter (Lu) ^[1]
Bidirectional Class:	Left To Right (L) ^[1]
Combining Class:	Not Reordered (0) ^[1]
Character is Mirrored:	No ^[1]
GCGID:	LS220000 ^[5]
HTML Entity:	Š Š

Š

UTF-8 Encoding: 0xC5 0xA0

Copy the Š into notepad ++ twice
 Use the plugin converter to convert the second from ASCII to HEX



This confirms the original UTF8 had a hex value of C5A0 = UTF-8 Encoding from compart website.

The compart website shows the escape sequence required is Š

ONIXCHR

The ONIXCHR record to ap the S with caron is as below

```

XAW005G          Control File Maintenance          3/20/23 1

Application : ONIXAIDA ONIX USING AIDA EDI
Key . . . . : ONIXCHR Mapping for nonstandard characters
Maximum rec : 999 Last change : 3/19/23 IPTORMC L1:
Allow Dup . : 2 1-Yes/2-No L2:
Usage . . . . : Pos to . .
                An item
                with this
                character UTF8
Code | Value Description character UTF8
CCCCC S VVVVVVVVVVDDDDDDDDDDDDDDDDDDDD IIIIIIIIIIIII UUUUUU
6722 &#322; l with stroke 9781772840469 C582
6741 &#352; S with caron 9780806155142 C5A0
  
```