

ISTC Maintenance Guide

Version History

Version	Author / Contributor	Date	Comments
1	Catherine Smith	18/11/09	Text File
1.1	John Harrison	29/04/2010	Original Text Imported Troubleshooting Key(s) Added

Table of Contents

Overview.....	2
Packages Required.....	2
Interfaces.....	2
Installing TTF on linux.....	3
Maintenance.....	3
The HTML folder - An Explanation.....	3
The XSLT - Some potentially useful remarks.....	3
Searching and Index Background.....	4
The Monthly Cron Job on the server.....	5
Instructions For Tasks You Might Be Asked To Perform.....	5
Add locations from a new country to editor and display.....	5
Change the German data	5
Change the links to different catalogues from the record display.....	6
Print the full ISTC to pdf.....	6
Recreate USA data from files.....	6
Recreate Bibliographical data from files.....	6
Clear the editing store.....	6
Cengage Data Request.....	6
Other scripts that might be useful in case the same or similar requests come up again.....	7
Appendix A – Troubleshooting Keys.....	8

Overview

Database holding ISTC records and hosted on behalf of the British Library (istc.bl.uk). Our BL contact is John Goldfinch (john.goldfinch@bl.uk)

The application consists of three Cheshire3 databases.

1. The main database (id = db_istc) holds ISTC bibliographical records in MARC21XML.
2. The Bibliographical database (id = db_refs) holds key value pair style data for short bibliographical codes used in field 510 of the MARC records and their full details. Not all 510 entries are the short version. Rarely used bibliographical references will occur in full in the MARC record and not appear in this database. This mainly makes display and search a little tricky.
3. The USA Locations database (id = db_usa) holds key value pair style data for USA location codes and their full details. In this case ALL USA location entries occur in the MARC records using the code.

Packages Required.

The application needs base, web, sql and formats packages from Cheshire3 as well as the special ISTCextras folder which contains a few extensions and specialist normalizers etc. It also needs True Type Fonts installed on the server in order for the pdf output options in the admin menu to work (instructions for installing TTF on linux are below).

Interfaces.

There is a search interface open to all, an editing interface to be used by contributors for editing and creating ISTC records and an admin interface used by British Library staff only.

Access to the editing interface is controlled by istcAuthStore and new users can be added by BL staff via the admin interface (create general user). If requested they can also be added using the run script using the flag -adduser

Access to the admin interface is controlled by istcSuperAuthStore and new users can be added by BL staff via the admin interface (create Administrative user). If requested they can also be added using the run script using the flag -addsuperuser

The SRU interface is turned on and has a filter in place so that private location data is not provided.
Filter = sruOutgoingTxr / sruOut.xml

python run.py -load deals with all three databases. There are individual flags too if you want to do them separately.

Installing TTF on linux

For Fedora 9+ which no longer has the rpm for chkfontpath used by the straight sourceforge download (<http://corefonts.sourceforge.net/msttcorefonts-2.0-1.1.spec>)

The version here has removed the need for that file.

as cheshire

```
# wget http://pfields.fedorapeople.org/packages/SPECS/msttcorefonts-2.0-1.1.spec
```

```
# rpmbuild -bb msttcorefonts-2.0-1.1.spec
```

as root

```
# rpm -ivh /home/cheshire/rpmbuild/RPMS/noarch/msttcorefonts-2.0-1.1.noarch.rpm
```

at the second stage you may need to install cabextract and ttmkfdir

```
# yum install cabextract
```

```
# yum install ttmkfdir
```

Maintenance

The HTML folder - An Explanation

Because of the restrictions of the BL style guide there are lots and lots of html pages which make up the basic page structure.

There are two main template files:

1 - baseTemplate.html

2 - printTemplate.html

These contain the basic header and footer stuff (one in colour one B&W). Each has a place holder for content and the colour one (because it is used for everything for display on screen) also has a place holder for navigation.

All the *Nav.html files in the html folder do the navigation and breadcrumb trail for the various different main pages.

The remainder are various menu's and other files read by the handlers to create the pages needed.

The XSLT - Some potentially useful remarks

There are a few generally useful XSL/Cheshire3 Transformers here which you might need if you are asked to write other scripts for the ISTC.

indentingTxr/reindent.xsl - properly indents the XML - useful for all writing out to file

filterTxr/filterMarcFields.xsl - filters out the fields usually required when supplying data for data

requests

toTextTsr/toMARCTxt.xsl - produces Marc Aleph output as text for printing etc
(toAlephTsr/toAleph.xsl does the same but in html for screen display)

dataTransformer/toMarcXml.xsl - is not used any more but was used to turn the data from Cheshire2s marc record format to MARC21XML may be useful in other projects moving marc records from Cheshire2 to Cheshire3 so kept for that reason only.

recordDisplay.xsl - This is a very complex XSLT because it is the basis of four different transformers which deal with the various methods of outputting the data from the search screen, the screen, email, print and save. Theoretically no changes will be needed to this file but at least if they are you will only have to make them once even if the structure is a bit complicated to figure out at first.

The nasty looking text in this file is actually the set up for the RTF output file used for emailing and saving. The code comes from the template called outputTemplate.txt in the www/istc folder. This is a file created in RTF and then saved as txt so that you can find the header and footer and all the stuff that formats each entry. Its not pretty I know but it does do the job. No changes should be required but if they are then basically you make the changes to the rtf using open office and a few records and then cut and paste the formatting stuff from around the data into the xslt at the relevant points. Any changes at all usually mean copying everything back in to the xslt even if its really minor because at the beginning seems always to be affected.

Bibliographical references (the full version) and USA locs are added in the python handler code and replace place holders set up in the xslt.

printAll.xsl is similar to recordDisplay but deals with the transformation of the record in reportLabDocumentFactory for the pdf output options available from the admin menu. It is a much simpler file and rather than using a placeholder for usa locations it retrieves them live from the xslt. I tried this for the recordDisplay on the screen but it cannot return the document quick enough to be used on screen kept getting python errors. Here it is necessary because the transformation has to happen in a single process without using Python as it is actually part of a document factory.

Searching and Index Background

Generally straight forward there is a zeerex mapping.

Slight oddities with use of pass-through indexes and some which are searched in pairs all described below.

USA locations use regular pass through indexes.

Biblical references indexes use special ISTC extension pass through indexes because of the special structure needed for returning the data (due to spaces in the codes).

norzig.posessingInstitution (locations) searches general locations and the USA locations index

istc.referencedBy (bibliographical references) searches both the abbreviations and the full references

The Monthly Cron Job on the server

The cron job is programmed to run using crontab to run once a month and the script itself is here:

/home/cheshire/copyISTClogs.sh

This copies the log files for the previous month to the BritLib user for the BL webteam to download and also backs up the data to scabucks.

Instructions For Tasks You Might Be Asked To Perform

Add locations from a new country to editor and display

1. Add the data to the records (dataCleaning/addAustrian.py is an example of how this was done for Austrian data)
2. Add new field to the following indexes (configs.d/indexes.xml)
 - a) idx-kwd-location
 - b) idx-location
 - c) idx-location-private
3. Add to the following xsl stylesheets
 - a) recordDisplay.xsl -- copy version for another country and change the tag number the country name and all variables of the form v[number] and l[number] to the next unique number (note they are not necessarily numbered in the order they appear) -- also add to list of tags in the if statement in the first line of the 'locations' template
 - b) istcForm.xsl -- Inside <div id="holdingscontainer"> add to the list of <option> tags and then later there is a choose statement with all of the location tags listed the new tag needs adding here and the further down there are some method calls to accesspointlist so copy an existing one change the tags numbers. Further down is the template called accesspointstring here there are two long choose statements which need the new tag adding to the list.
4. add to the following javascript files
 - a) searchForm.js - add to locList array
5. in ISTCextensions in code directory add to dictionary in LocCountriesNormalizer
6. in dataRequest.html add to option list

Change the German data

This is a fairly regular job as the German contributors have a different system which provides an export of their location data for us to upload.

It has been agreed with John Goldfinch that all future exports of this data will be supplied to us in the form found in dataCleaning/newGermanData.txt.

The file should follow this layout exactly and should be in unicode. If not you can send it back to John Goldfinch and ask him to change it.

The script dataCleaning/changeGerman.py actually makes the changes to the data. There is one flag in the script which determines whether private data is to be kept or whether all German locations should

be deleted. keepPrivate = True will retain any locations marked as private set to False it will delete all of them. NB it only deletes German locations for those files which have new entries in the provided data. Before you do this make sure the editing store has been cleared (see below).

Change the links to different catalogues from the record display

BSB-Ink reference links are done in the python handler with the place holder being added by the xslt. This is because the URLs are not completely formulaic so the data is taken from the 530 u field if there is data there and if not follows a formular.

Other links (GW) are done in the recordDisplay xslt.

The links from the Koninklijke Bibliotheek notes are also done in the recordDisplay xslt.

Print the full ISTC to pdf

This takes a very long time during which you must not restart apache on the machine it is running on as the XSLT needs to access the server to pull in the US data. The script is data_requests/printToPDF.py

Recreate USA data from files

There is a script for this (dataCleaning/createUsaData.py) works on data in the format of dataCleaning/newusadata.csv.

Recreate Bibliographical data from files

There is a script for this (dataCleaning/createBiblioData.py) works on data in the format of dataCleaning/newBiblioData.txt

Clear the editing store

Sometimes they will want all the things currently in the editing store deleted. You will also want to do this before making any changes to the live data so that things in the editing store don't overwrite your changes.

```
python run.py -clearEditingStore
```

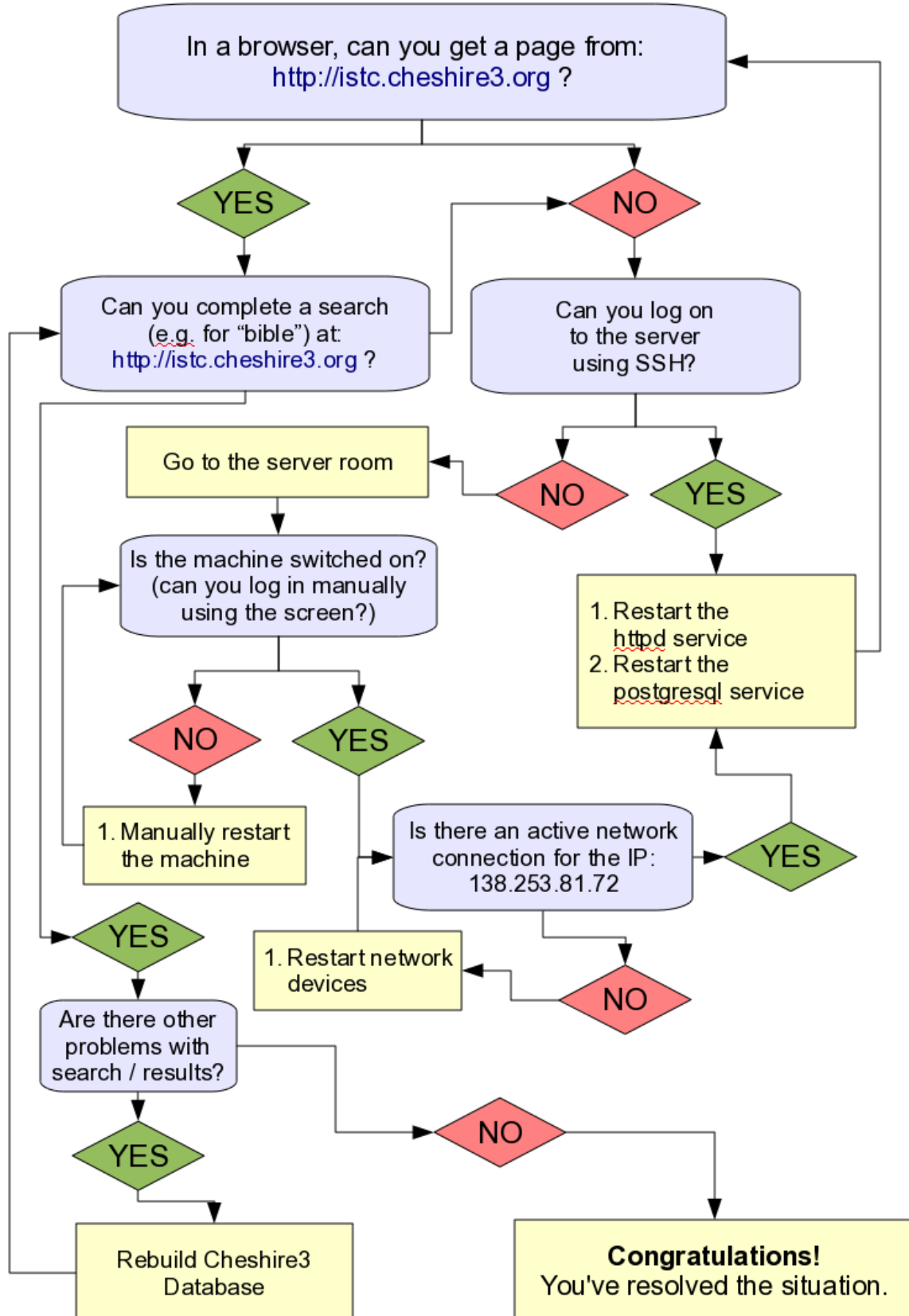
Cengage Data Request

I have sent an export sample of 20 records containing Cengage Learning Microfiche data to the BL for approval by Cengage. If John Goldfinch comes back saying they want to go ahead with all of the data the script is in data_requests/cengageData.py and will just need the 20 record restriction taken off and then should be ready to go.

Other scripts that might be useful in case the same or similar requests come up again

1. `bsbConcordance.py` - creates a list of BSB-Ink numbers matched with their corresponding ISTC numbers.
2. `germansinglelib.py` - prints out certain fields of all records which only have one German location which is not Munich ordered by location and then ISTC

Appendix A – Troubleshooting Keys



Key 1: Web Accessibility