

# The Bodleian Library *allegro* catalogue of Japanese books – using NACSIS data in a western library system<sup>1</sup>

DAVID HELLIWELL

---

## Summary

Since the early nineties, the Japanese librarians in the Bodleian Library have been participating in a collaborative effort to make a UK union catalogue of Japanese books hosted by NACSIS. Cataloguing is done online, and the downloaded records of Oxford holdings (bibliographical records and copy records) are then converted into a format that can be stored, indexed, accessed and displayed by *allegro* in conjunction with a suitable FEP (front-end processor).

The presentation will show how these processes are effected, thereby introducing all the main features of the *allegro* software. The extreme differences in format between the original NACSIS data and its *allegro* equivalent, as well as how the data is handled, will illustrate the near total user-configurability of *allegro*. Finally, there will be an online demonstration of the web interface to the catalogue through the *avanti* server.

---

## Background

In 1991, the National Center for Science Information Systems (NACSIS) offered its services free of charge to a consortium of UK libraries with the aim of facilitating the cataloguing Japanese material and thus contributing towards the formation of a worldwide union catalogue of Japanese books.<sup>2</sup> The UK holdings were first made available to readers by means of a CD-ROM issued quarterly, and more recently, online from a fileserver in Cambridge.

---

<sup>1</sup> This paper was presented at the conference of the European Association of Japanese Resource Specialists (EAJRS) in Leuven in September 1998. A reduced version was published in *Program* 33:3(1999).

<sup>2</sup> Brian Perry, in 学術情報センター紀要 第8号(1996), 385-393.

In Oxford, Japanese software is dispensed with entirely, and the data presented by means of *allegro*, the German library database system developed by Bernhard Eversberg at Braunschweig Technical University Library.<sup>1</sup> The import (or conversion) of the original NACSIS data into an *allegro* format, the construction and indexing of the database, the incorporation of subsequent additions and emendations, and the simultaneous delivery of the live data through DOS, Unix (including WWW) and Windows interfaces not only provides an illustration of all the principal *allegro* functions, but also demonstrates their most important characteristic, a near total configurability to the requirements of the user.

### Structure of NACSIS data

The download data produced by NACPC, the software used for online NACSIS cataloguing, is a subset of the so-called “B-format”<sup>2</sup>, which is best illustrated by the following example taken from a recent Oxford download file:

#### Example

```
<BA3609710X>
GMD:
SMD:
YEAR:1998
CNTRY:ja
TTLL:jpn
TXTL:jpn
ORGL:
VOL:上
ISBN:4625513138
PRICE:5200円
OTHN:JLA:98025743
TR:万代和歌集/安田徳子-||マンガイワカシュウ
PUB:東京:明治書院,1998.6
PHYS:iv,367p;22cm
NOTE:監修:久保田淳
NOTE:底本:龍門文庫(奈良県吉野郡所在)所蔵本
PTBL:和歌文学大系||ワカブンガクタイケイ<BA31455304>13//b
AL:安田,徳子(1946-)||ヤスダ,ノリコ<DA10217563>
CLS:NDC8:911.108
CLS:NDC9:911.108
SH:NDLSH:和歌--評釈||ワカ--ヒョウシャク//K
SH:NDLSH:万代和歌集||マンガイワカシュウ//K
****ENDOFDATA****(B202E)[NC]Mon Aug 17 14:27:52 1998
<BA3609710X>万代和歌集/安田徳子-;上.--明治書院,1998.--(和歌文学大系;13).
<FA012965>
```

<sup>1</sup> Described by Eversberg in *Program* 29:2 (April 1995), 147-158.

<sup>2</sup> Set out in appendixes 6 and 7 of the NACSIS users' handbook (目録所在情報サービス利用の手引, 1986 edn., 161-184).

OXFORD/  
<CC0244341088>  
LOC:BJL  
VOL:上  
CLN:Jap. e. 17386/13  
RGTN:  
CPYR:  
CPYNT:  
LDF:  
LTR:  
\*\*\*\*ENDOFDATA\*\*\*\*(B203N)[ ]Mon Aug 17 14:28:18 1998

The main characteristics of this data relevant to the process of converting it into an *allegro* format are as follows:

- Its structure is regular. The string “ \*\*\*\* END OF DATA \*\*\*\* ” provides a unique end-of-record marker, and the fields all start on a new line.
- Bibliographical records and copy records are separate, but the bibliographical record number (in this case “ < BA3609710X > ”) appears in both, and is the first piece of data in each record.
- The format is simple and transparent, especially to speakers of English, but is somewhat redundant, as many of the possible field tags are present even if the field contains no data.

### **Conversion into *allegro* format**

Essentially, the process of conversion consists of two simple processes: mapping data from one field to another, cutting it up or combining it as necessary, and converting the data from one form to another. This is done by the program *import*, which produces a datafile that *allegro* is capable of turning into a database. Such datafiles are called “basic” *allegro* files, and have the following structure:

- Each record occupies a single line, separated from the next record by a carriage-return and line-feed (ASCII 13 10).
- The data is organised into fields, each introduced by a tag that is usually 2 or 3 bytes long and begins with a number.
- Each field is terminated by ASCII 0.<sup>1</sup>

Like all the other programs that constitute the *allegro* package, how *import* functions is determined by so-called “parameter” files that are written by the user in a language peculiar to *allegro*. The language is

---

<sup>1</sup>For purposes of illustration however, *allegro* (like MARC) records are usually set out with each field on its own line. Also, the field tag is prefixed with a hash # sign, but this is not stored in the datafile.

defined in the *Systemhandbuch*<sup>1</sup> and is constructed to give the user as much freedom as possible in defining the structure of the database.

The basic parameter is the configuration file, which determines the basic shape of the data, such as the field tags that will be accepted, what subfields (if any) they may contain, and whether fields or subfields may be repeated. It also provides for a number of checking routines that can be applied to the data, such as the verification of ISBNs and the marking of initial articles in titles. A configuration file is required by all *allegro* programs with the exception of a few relatively modest utilities, and is designated by a single letter of the alphabet. That used in Oxford for the CJK data is L, which is also the name of the format that has been devised for it.<sup>2</sup>

Other parameter files are required for exclusive use by specific programs, and the import parameter is the one in which the conversion routines are defined. To do its work, *import* must first be told the shape of the data to be converted, and individual portions of it must then be identified and converted into the *allegro* format described in the configuration file.

A couple of import routines from the above examples will suffice to give the flavour of the *allegro* import language, and also to illustrate the extent and nature of the conversion from NACSIS download format to Oxford's *allegro* L format.

The routine (called a "paragraph") that imports the country code is

```
#0cc  
s "CNTRY: "
```

It tells the program to look for the string beginning "CNTRY:" in the foreign data, and to put what is found into *allegro* field #0cc. The conversion is therefore from

```
CNTRY: j a  
to  
#0cc j a
```

---

<sup>1</sup>Bernhard Eversberg: *allegro-C Systemhandbuch, Version 15* (Universitätsbibliothek der TU Braunschweig, September 1997), ISBN 3-927115-33-9.

<sup>2</sup> The reasons why an existing format has not been used for this purpose are not significant here, nor ultimately in other contexts either. In the CJK (Chinese/Japanese/Korean) area, it is not differences of format that are preventing shared cataloguing, but fundamental differences in character encoding and romanisation (including syllable aggregation), as well as the absence of an agreed system of subject headings. The argument should centre not on the standardisation of bibliographical formats, but in what form data should be entered into those formats.

Other processes are much more complicated, and are often best done in stages. The following routine is the first part of the paragraph that imports the main series title:

```
#560
s"PTBL: "
e"<"
_ "||"
_ "3k"
```

The `s` command is the same as those in the previous example, but in this case we must stop importing the data at the string “<”, and then change the string “||” into subfield `3k`, which contains a *katakana* transcription of the data in the main part of the field. The state of the conversion at this point is from

```
PTBL:和歌文学大系||ワカブンガクタイケイ<BA31455304>13//b
```

to

```
#560 和歌文学大系3kワカブンガクタイケイ
```

But in western libraries, romanisation is required, and this can be generated automatically from the *katakana* transcription. Another paragraph does this, and contains a long set of substitutions that converts the *katakana* into romanisation, which is then put into subfield `3r`. It is worth taking a closer look at the *katakana* substitution process, as though essentially simple, it is very extensive, with over 400 pairs of substitutions, and generates a vast quantity of data which under normal circumstances would have to be entered manually. Furthermore, it can even take account of common mistakes in transcription, so that the resulting romanisation actually provides more accurate indexes than the *katakana* original.

The reason why over 400 pairs of substitutions are required to transcribe only 70 or so *katakana* syllables is because in combination, the sounds change. For example, *shi + yo* produces not *shiyō*, but *shō*. It is necessary to take account of all such combinations before the individual syllables can be converted, so the substitution routine must begin with the longest combinations:

```
_ "ッキャー"
_ "kkyaa"
_ "ッギャー"
_ "ggyaa"
_ "ッキュー"
_ "kkyuu"
```

and then proceed through the shorter ones, to the final single syllable substitutions, ending with “n”, the only final consonant in Japanese:

\_"ワ"  
\_"wa"  
\_"ヲ"  
\_"o"  
\_"ン"  
\_"n"

The numbering within the series is then imported into subfield 3n so that the final result of the conversion is

#560 和歌文学大系3kワカブンガクタイケイ3rwaka bungaku taikei3n13

Here is the complete result of the conversion process applied to the two NACSIS download records quoted above:

### *bibliographical record*

#000 BA3609710X  
#0ca x1998  
#0cc ja3tjpn  
#100 4-625-51313-83v上  
#250 万代和歌集3kマンダイワカシュウ3rmandai wakashuu  
#290 安田徳子  
#340 安田, 徳子 (1946-)3kヤスダ, ノリコ3ryasuda, noriko  
#406 東京 : 明治書院, 1998.6  
#470 iv, 367p ; 22cm  
#560 和歌文学大系3kワカブンガクタイケイ3rwaka bungaku taikei3n13  
#600 監修: 久保田淳  
#600a底本: 龍門文庫(奈良県吉野郡所在)所蔵本  
#705 911.108  
#760 和歌 -- 評釈3kワカ -- ヒョウシャク3rwaka -- hyoushaku\*k  
#760a万代和歌集3kマンダイワカシュウ3rmandaiwakashuu\*k

### *copy record*

#000 BA3609710X  
#805 Jap.e.17386/133v上

The import process has been considered at some length because it is the key to the successful formation of a database completely independent of the Japanese software that usually handles NACIS data. It remains to examine how the database is constructed and maintained.

## **Constructing the database**

Once a “basic” *allegro* file has been produced by the conversion process effected by *import* as defined in the configuration file and import parameter, it is the task of other programs in the *allegro* package to turn it into a database. Rather than keep the copy records separate and retrieve them on export, as is done in very large databases where there may be

dozens or even hundreds of such records linked to a single bibliographical record, in the Oxford catalogue they are integrated.

First, a database is made from the bibliographical records only with the program *index*, which produces a set of index keys defined by the index parameter – again, entirely user-configurable. The program *update* then takes the copy records and integrates them with the bibliographical records. This program depends on the presence in each record of a *Primärschlüssel*, or “primary key” – a unique record identifier, which in L data is found in field #000. It takes a copy record and looks for the primary key in the database index. If it is found, the fields containing the copy information are added to the bibliographical record, and any additional index entries are produced, again as defined in the index parameter. And so, to take the above example once again, we end up with a single record in the database:

```
#000 BA3609710X
#0ca x1998
#0cc ja3tjpn
#100 4-625-51313-83v上
#250 万代和歌集3kマンダイワカシュウ3rmandai wakashuu
#290 安田徳子-
#340 安田, 徳子 (1946-)3kヤスダ, ノリコ3ryasuda, noriko
#406 東京 : 明治書院, 1998.6
#470 iv, 367p ; 22cm
#560 和歌文学大系3kワカブンガクタイケイ3rwaka bungaku taikai3n13
#600 監修: 久保田淳
#600a底本: 龍門文庫(奈良県吉野郡所在)所蔵本
#705 911.108
#760 和歌 -- 評釈3kワカ -- ヒョウシャク3rwaka -- hyoushaku*k
#760a万代和歌集3kマンダイワカシュウ3rmandaiwakashuu*k
#805 Jap.e.17386/133v上
```

It is worth taking a look at what index entries we have caused the programs *index* and *update* to produce from the above data. In the following table, after the initial bar, the index number is given (0 being the “short-title list” that is displayed when the “show titles” button is used) followed by the index entry. The form and content of each index entry is determined by the index parameter, which can be varied at will.

0	万代和歌集	安田徳子-	1998	short-title list
1	mandai wakashuu			titles (including series)
1	waka bungaku taikai ;		13	
1	万代和歌集			
1	和歌文学大系 ;		13	
2	mandai			words in titles (romanisation)
2	wakashuu			
2	評釈			words in subject headings ( <i>kanji</i> )

2万代和歌集	
2和歌	
3yasuda, noriko 安田, 徳子 (1946-)	authors
3安田, 徳子 (1946-) yasuda, noriko	
5明治書院	publishers ( <i>kanji</i> only)
64-625-51313-8	standard numbers (ISBN, ISSN)
7911.108	classification (NDC8)
7万代和歌集	subject headings (NDL)
7和歌 -- 評釈	
8jap.e.17386/13	shelfmarks
9ba3609710x	record numbers

## Working routines

There are both DOS and UNIX versions of the *allegro* programs. Although there are also Windows and WWW interfaces, most *allegro* operators prefer to work with the original DOS versions, which permit much faster handling and processing of the data. They are activated with command lines which may be either entered into the “target” box of a Windows “shortcut”, or incorporated into a batch file. In Oxford, the various stages of the downloading routine are incorporated into a batch file and presented to the operator through the following screen:

```

      NACSIS IMPORT MENU

F1  Download records (on diskette) ==> EUC file
F2  NACSIS bib. records == > tagged (PK) format
F3  NACSIS copy records == > tagged (PK) format
F4  NACSIS records ==> temporary database
F5  View temporary database
F6  Temporary database ==> main (OXJ) database
F7  View main (OXJ) database
F8  Backup main database (OXJ) logfile to diskette
F9  Total backup of main database to diskettes
F10 Quit to DOS

```

The operations are presented in the order in which they are normally used, the working routines having been carefully planned.



F1

The downloaded NACSIS records are taken from the NACPC machine on a diskette, which is inserted into the local workstation. They are converted from Shift-JIS back to EUC coding (before downloading, in NACSIS “B-format” they are encoded in EUC!) with a conversion utility.

F2

The bibliographical records are converted into tagged *allegro* format (as shown above) with *import* and viewed, so that the operator can make sure that the import process has run correctly.

F3

The same, but for the copy records.

F4

Now the downloaded records are imported into “basic” *allegro* files (whose structure is described above), the bibliographical records are made into a “temporary” database (any existing temporary database having been deleted), and finally the copy records are integrated. This operation uses the programs *import*, *index* and *update*.

F5

The temporary database is accessed with the principal *allegro* program *presto*, which also allows the operator to edit the records. This is done to ensure that the structure of the records is sound, and to enable any minor modifications to be done before the records are finally introduced into the main online catalogue.

F6

Once the operator is satisfied that the temporary database is in order, the records are exported from it with *srch*, the program which does full-text searching and export, and integrated with the main catalogue using *update* again. This time, *update* is started up not in a mode which merges fields, but which completely overwrites the old record if it already exists, or creates a new one if it doesn't. In this way, any records which may have been revised on NACSIS and downloaded for a second time will not create duplicate entries in the database. At the same time, the date of the update is automatically entered into field #0ru of the record.

F7

The main catalogue is viewed with *presto*.

F8/F9/F10

These functions are self-explanatory. F8 and F9 are no longer necessary as the data on the fileserver is automatically backed up each night; in any case, the catalogue is now far too big to fit on to diskettes.

## **The OPACs**

Describing the user interfaces to *allegro* is another matter, but by way of conclusion I append some illustrations of how the converted NACSIS data (again using the above example) is eventually presented to the reader. There are two ways of gaining access to the catalogue, and both access the same, live datafiles.

Locally, the DOS version is preferred, as it does not need a mouse and is therefore much easier to manipulate. It also reacts more quickly – in fact, almost instantaneously – thus enabling multiple searches to be done extremely efficiently. The DOS version uses the FEP JitsuYou, which was specially commissioned by the Bodleian Library and the British Library for this application.

Remote (as well as local) users may use the WWW interface.<sup>1</sup> We recommend the use of UnionWay as an FEP.

---

<sup>1</sup> Address: <http://www.bodleian.ox.ac.uk/dept/oriental>. There is also telnet access ([erl.ox.ac.uk](telnet://erl.ox.ac.uk)), but as many users have experienced problems with configuration and the compatibility of FEP's, we are considering discontinuing it in favour of the more recently developed web interface.