

Appendix 3: Additional ISNI Search Options

Changing the Encoding View

Searches within the WinIBW interface operate on the index keys of each record; these are a distinct set of fields separate from the Marc and PICA codings. It is sometimes useful to view records by these keys to better understand the results that your searches will produce. To do this:

- 1) Navigate to any record
- 2) Type into the command line "sho key".

To return to the Marc or PICA view, enter "show m21" or "sho p", respectively. The view will remain in the new format until it is changed back manually or WinIBW is restarted.

NOTE: Search syntax in WinIBW is **not** case-sensitive; "SHO KEY" works as well as "sho key", as does "Sho key".

Search Fundamentals

Searches in WinIBW operate on keys. For example, to search for authors with the surname Pratchett, we would enter:

"FIN NA: Pratchett" or "SCA NA: Pratchett"

into the command line. FIN finds all records matching this name, and SCA performs a browse search for this name, starting with either a record matching the search or the first subsequent record alphabetically; A search "FIN TI: War and Peace" would first bring up "War and Peace" as the first result; the search term does not exist as a record, so the next record alphabetically is displayed first.

Search term	Function
FIN	'Find' – A relevance-based search
SCA	'Scan' – Browse for the search term alphabetically.

In general, all search operators can be utilised in any combination, though care must be taken that you mean what you say, and say what you mean!

Keys

Searches can be conducted on any key in the database; these include common choices such as "AW" for a basic 'any word' keyword search, "NW" for a 'name word' keyword search in the 'name' field, and "NA", a non-keyword based search of the 'name' field.

There are two types of keys, keyword keys such as "NW" – 'name word' – and non-keyword keys such as "NA" - name . Search terms entered after a keyword

key may be entered in any order, while terms entered after a non-keyword key must be entered in the order they appear in the results¹.

Boolean Operators

AND (&), OR (|), and NOT (-) can be used to restrict a search in the command line. For example, to search for the (exact) title "Mort" and the (exact) author "Pratchett, Terry" we would enter into the command line:

```
FIN NA: Pratchett, Terry AND TI: Mort
```

Possible alternatives would include "SCA NA: Pratchett, Terry AND TI: Mort" and "SCA (NA: Pratchett OR NW: Terry) AND TI: Mort".

Mandatory Terms

Preceding a search term with "+", e.g. "SCA TW: Walking on +Ice" will return results from "Walking On Ice" onwards in the database, as long as they contain the term "ice". This is the opposite of the "NOT", or "-" operator.

Operator Precedence

The order in which operators are applied is important, and their order of application is shown below:

Order:	FIRST	SECOND	THIRD	FOURTH
Operator	Wildcards: ~ ! # % ? *	<default operator>	AND	OR

So, "FIN AW: Hannibal OR Montgomery, Charles AND war#" is equivalent to searching for "Hannibal OR [Montgomery, Charles AND [great [war#]]]".

"Special words" and brackets

Some words are interpreted by WinIBW in a special way when searching. These include the key field abbreviations, such as "AW" and "NA", but also words such as "AND", "or", "to", "in" and "for". This is to improve usability in searching and save processing power on the database servers. This can be seen in the example of searching for "War and Peace"; the results produced would be the same as for "War Peace", and this is because the word "and" is interpreted as the operator "AND", and because two search terms appearing next to each other are searched using "AND" by default.

¹ Unless a search operator such as "%" is used that tells the search to ignore order.

To prevent these words being specially interpreted, they can be preceded by the backslash character: "and" → "\and".

Quotation marks and exact searches

Different quotation marks do different things in WinIBW searches. The backslash, "\", as mentioned above, is a special kind of quotation mark that only affects the character immediately following it. It does not affect anything else such as word order requirements, etc.

Double quotes, e.g. "\" interpret special words as normal words, but do not affect symbols such as "&" or "+". Words within the quotation marks must be in the correct order in any records found.

Backquotes, e.g. "`Schrodinger`" will find all instances of the included word with accents exactly as presented, - in this case, "Schrödinger" will not be found.

Single quotes, e.g. "'War'" will find instances of the word with any distribution of accents if no accents are specified, but if accents are specified, e.g. "FIN AW: 'Schrödinger'" only results with matching accents will be found.

Apostrophes will not be interpreted as quote marks if they are embedded within a word: "O'neil's" would be interpreted as a single text string, not "O AND 'neil' AND s".

Brackets

Brackets can be used to change the hierarchy of search terms and group search terms together; in the final Boolean example above, brackets are used to search for records containing the title "Mort" that had either an exact name match with "Pratchett" or a name keyword "Terry". If the brackets were omitted, the search "SCA NA: Pratchett OR NW: Terry AND TI: Mort" would produce records that either had both the name keyword "Terry" and the title "Mort", or had just the name "Pratchett".

Searches Using Previous Results

Chapter 2 has already introduced the *history* tab within the WinIBW search results. It is possible to use the searches listed here to refine subsequent search results. If I first searched for:

```
FIN NA: Pratchett, Terry
```

and then clicked the *history* button on the search results page, this search would be displayed on the screen as:

```
S1 FIN NA: Pratchett, Terry .
```

I could use this to refine a subsequent search for the title "Mort" by combining the two as follows:

`FIN s1 AND TI: Mort .`

This is equivalent to typing in "`FIN NA: Pratchett, Terry AND TI: Mort`". This facility is most useful to find the intersection of several previous searches, and is compatible with all other search options listed in this appendix.

Wildcard Characters

Wildcard characters can be used within words in the search term to search more broadly. Available wildcard characters are as follows:

Character	Use	Example
? or *	Replaces any number of characters.	use* will find: use, used, user, user-friendly
#	Replaces 0 or 1 characters	use# will find: use, user, used, NOT user-friendly
!	Replaces 1 character	organi!ation will find: organisation, organization

Wildcard characters can **not** be used at the beginning of search terms, e.g. "`FIN TW: ?ing Bear`" would return no results. They can be used in the middle or at the end. "`Use?`" would find both "User-friendly" and "User Troubleshooting", as the "?" wildcard is at the end of a word.

Multiple wildcard characters can be used within one search term, e.g. "`organi!ation?`"

Approximate Searching

It is possible to search for variations on a search term by combining it with the "/n" operator. Searches using this operator will find any results that can be made to match the word to which the "/n" operator is appended by adding, removing or changing n letters. For example:

`FIN NA: Jackson/3`

would return any results with a name matching "Jackson", "Jack", "Blackson", "Jacksons", "Blackton" or "Jacko", but not "Jillsons".

Proximity Operators

Just as wildcard characters can be used within a word to search for variations, the same characters can be used as Proximity Operators when not attached to another search term to represent whole words:

Character	Use	Examples
-----------	-----	----------

? or *	Replaces any number of words	"jnam ? proq" will find: "jnam proq", "jnam viaf proq" and "jnam viaf viaf proq", but not "proq jnam"
#	Replaces 0 or 1 words	"jnam # proq" will find: "jnam proq" and "jnam viaf proq", not "jnam viaf viaf proq" or "proq jnam"
!	Replaces 1 character	"jnam ! proq" will find: "jnam viaf proq", not "jnam proq" or "jnam viaf viaf proq"

Proximity characters can be combined in the same way as any other operator, e.g. "Maxwell ##! Boltzmann" will return results in which "Maxwell" and "Boltzmann" are separated by 1, 2 or 3 intervening terms, but not 4 or 0.

They can also be combined with a "/n" operator, e.g. "FIN TI: Alice? #/2 Wonderland" will find both "Alice in Wonderland" and "Alice's adventures in Wonderland".

Searching In Any Order

The "%" operator when used in a search will allow search terms to be found in any order: "FIN TW: Fire % Fighting" will find all the results for both "FIN TW: Fire Fighting" and "FIN TW: Fighting Fire". This can be combined with other operators as usual, e.g. "FIN AW: Willows %? Wind" will return any records with a field containing both the terms "Willows" and "Wind", in any order at any separation.

The NEAR and NEAR/n Operators

The "NEAR" operator searches for words (in any order) separated by up to two other search terms: "FIN AW: Einstein NEAR Albert" will find "Albert Einstein" and "Charles Einstein Julian Ray Albert", but not "My cat Albert likes to think that he's Einstein" (however unlikely a title that may be).

The near operator can be modified to the form: "NEAR/n", for which n is any number. This will find any results separated by n words; in comparison with the above example, "FIN AW: Einstein NEAR/5 Albert" would find "My cat Albert likes to think that he's Einstein" in addition to the other results.