

# Problems

## pageInfo should belong to citations, not citation groups

- Creating `[click:Separate]` revealed a `(problem)`: pageInfo strings are conceptually associated with citations, but the implementation associates them to citation groups.  
The number of available pageInfo slots changes during `[click:Merge]` and `[click:Separate]` while the number of citations remains fixed.
- The proposed solution was to change the association.
  - Not only reference marks (citation groups) need unique identifiers, but also citations. Possible encoding for reference mark names:  
`JR_cite{type}_{number1}_{citationKey1},{number2}_{citationKey2}`  
where `{type}` encodes the citation type (for the group), `{citationKey1}` is made unique by choosing an appropriate number for `{number1}`  
This would allow `JR_cite_{number1}_{citationKey1}` to be used as a property name for storing the pageInfo.

Changes required to

- reference mark search, name generation and parsing
- name generation and parsing for properties storing pageInfo values
- in-memory representation
  - JabRef 5.2 does not collect pageInfo values, accesses only when needed. So it would be change to code accessing them.
  - The proposed representation does collect, to allow separation of getting from the document and processing
- insertion of pageInfo into citation markers: JabRef 5.2 injects a single pageInfo before the closing parenthesis, now we need to handle several values
- `[click:Manage citations]` should work on citations, not citation groups.

## Backend

The choice of how do we represent the data and the citation marks in the document has consequences on usability.

Reference marks have some features that make it easy to mess up citations in a document

- They are **not visible** by default, the user is not aware of their boundaries  
(`L0:[key:Ctrl-F8]`, `L0:[View]/[Field shadings]` helps)
- They are **not atomic**:
  - the user can edit the content. This will be lost on `[click:Update]`  
If an `As character` or `To character` anchor is inserted, the corresponding frame or footnote is deleted.
  - by pressing Enter within, the user can break a reference mark into two parts.  
The second part is now outside the reference mark: `[click:Update]` will leave it as is, and replace the first part with the full text for the citation mark.
  - If the space separating to citation marks is deleted, the user cannot reliably type between the marks.  
The text typed usually becomes part of one of the marks. No visual clue as to which one.  
Note: `[click:Merge]` then `[click:Separate]` adds a single space between. The user can position the cursor before or after it. In either case the cursor is on a boundary: it is not clear if it is in or out of a reference mark.  
Special case: a reference mark at the start or end of a paragraph: the cursor is usually considered to be within at the corresponding edge.
- (good) They can be moved (Ctrl-X, Ctrl-V)
- They cannot be copied. (Ctrl-C, Ctrl-V) copies the text without the reference mark.
- Reference marks are lost if the document is saved as docx.
- I know of no way to insert text into an empty text range denoted by a reference mark
  - JabRef 5.3 recreates the reference mark (using [insertReferenceMark](#)) [here](#)
  - (change) I preferred to (try to) avoid this: [NamedRangeReferenceMark.nrGetFillCursor](#) returns a cursor between two invisible spaces, to provide the caller a location it can safely write some text. [NamedRangeReferenceMark.nrCleanFillCursor](#) removes these invisible spaces unless the content would become empty or a single character. By keeping the content at least two characters, we avoid the ambiguity at the edges: a cursor positioned between two characters inside is always within the reference mark. (At the edges it may or may not be inside.)
- (change) `[click:Cite]` at reference mark edges: [safeInsertSpacesBetweenReferenceMarks](#) ensures the we are not inside, by starting two new paragraphs, inserting two spaces between them, then removing the new paragraph marks.
- (change) [guiActionInsertEntry](#) checks if the cursor is in a citation mark or the bibliography.
- (change) `[click:Update]` does an [exhaustive check](#) for overlaps between protected ranges (citation marks and bibliography). This can become slow if there are many citations.

It would be nice if we could have a backend with better properties. We probably need multiple backends for different purposes. This would be made easier if the backend were separated from the rest of the code. This would be the purpose of [logic/openoffice/backend](#).

# Undo

- JabRef 5.3 does not collect the effects of GUI actions on the document into larger Undo actions.  
This makes the Undo functionality of LO impractical.
- [\(change\)](#) collect the effects of GUI actions into large chunks: now a GUI action can be undone with a single click.
  - except the effect on pageInfo: that is stored at the document level and is not restored by Undo.

## Block screen refresh

- LibreOffice has support in [XModel](#) to “suspend some notifications to the controllers which are used for display updates.”
  - [\(change\)](#) Now we are using this facility.
-