



Chapter 4

Known Limitations

The known limitations of the SoDA 3.0.2 release are described below. Where possible, work-arounds are also described.

Known Bugs

Filenames in Expressions must be Uppercase

When using filenames in And Where expressions, you must specify the pathname in all capital letters.

General Limitations

The following limitations apply to all SoDA users.

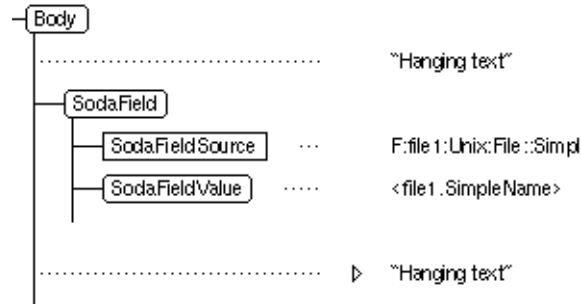
Visit Source

Visiting Frame Elements

Visit Source does not always work properly on elements in the Frame domain after documents are converted from Frame 5.1 to 5.5. Normally, this problem can be solved by simply regenerating the document.

Hanging Text

Hanging text is not handled well and, whenever practical, should be wrapped in a TextRange container. Hanging text is commonly created by the insertion of SoDA elements into the middle of paragraphs:



SoDA provides a utility for wrapping hanging text in a TextRange element. To run this utility on a document, choose the **SoDA:Utilities:Wrap Hanging Text** menu option. Note that the TextRange element must be defined within your EDD for the conversion to be successful (the SoDA default EDD defines a TextRange element).

Known problems with hanging text include:

- You cannot select a portion of the hanging text and then insert a field, replicator, or stipulator. A SoDA Error dialog box will appear saying that you have made an invalid selection.

For example, if you want to select some portion of hanging text and replace it with a SoDA field, you must instead delete the text and then insert the field at the cursor position.

- You cannot add a replicator or stipulator to a container that begins with hanging text, and if you try, a SoDA Error dialog box will appear that tells you so.

You can work-around the problem by wrapping that leading text in a container element.

Conditional Text in Master Elements

SoDA's use of conditional text to hide master elements (that is, elements with replicators attached) precludes the use of conditional text for other purposes within the master elements.

If you do tag text within a master element as conditional, those tags will be removed during document generation and will not appear in the elements added during generation. Once an element has been added, however, condition tags can be applied within it without being deleted by subsequent generations.

SodaHidden Tag Not Removed When Objects Reappear

If you have a SoDA document with a replicator over some objects, and you regenerate that document with the **Delete Mode** set to **Hide**, any elements that were previously generated for objects that have since disappeared will be hidden by the **SodaHidden** condition tag. Then, if you regenerate the document again, and some of those objects have since reappeared, the corresponding elements will remain hidden because the **SodaHidden** tag will not be removed from them.

Footnotes and Markers

SoDA elements in footnotes and markers are ignored—that is, never resolved—during document generation.

FrameMaker+SGML Files

When generating a book or any document in a book, SoDA may encounter various difficulties in opening a document. Such difficulties, which typically cause FrameMaker+SGML to display a dialog box if you try to open the document yourself, include:

- The presence of an autosave (*.**auto**) or recovery (*.**recover**) file. (If you open the document yourself, FrameMaker+SGML tells you that an autosave or recovery file exists and asks if you want to open it instead.)
If the **Ignore .recover and .auto Files** toggle button is set on the SoDA Generator dialog box, SoDA opens the (possibly outdated or corrupted) document, displays no message, and continues with document generation. The autosave or recovery file is ignored.
Otherwise, SoDA opens the autosave or recovery file, displays no message, and continues with document generation. The file is left open after generation so that you can decide how to save it.
- The presence of a lock (*.**lck**) file. (If you open the document yourself, FrameMaker+SGML displays a Document in Use dialog box.)
SoDA opens the document in View Only format, displays no message, and continues with document generation. SoDA will not be able to

Chapter 4 Known Limitations

generate the view-only document, but it will be able to use the view-only document for generating other documents (which could, for example, depend on global connectors in the view-only document).

Book files cannot be opened in View Only format, so opening a book that already has a lock file will cause an error.

- Missing fonts. (If you open the document yourself and it uses fonts that are not available on your system, FrameMaker+SGML displays an alert.)

SoDA opens the document despite the missing fonts, displays no message, and continues with document generation.

- Missing graphics. (If you open the document yourself and it imports graphics, by reference, that cannot be found, FrameMaker+SGML displays a Missing Graphic File dialog box.)

SoDA opens the document despite the missing graphics, displays no message, and continues with document generation.

You can avoid these situations by making sure that you can open all of the documents in your book without difficulty before generating them.

Replicators

Infinitely Recursive Replicators

It is possible to define recursive replicators that will cause SoDA to enter an infinite loop when you generate your document. For example, if you recurse down through the file system and encounter a symbolic link that points to a previously traversed parent directory, you will enter an infinite loop.

Unique-Only Sorting

The **Unique Only** toggle button in the SoDA Replicator dialog box is significant only when there is an expression in the **Ordered By** area.

Checking a Document Without an Insertion Point

If you do not have an insertion point in your document, **Check Document** will terminate without displaying the expected consistency report.

You can avoid this problem by clicking in your document, ensuring that you have an insertion point, before choosing **Check Document** from the **SoDA** menu.

Persistence in the Frame Domain

The Frame domain uses numeric object identifiers in references to Frame document elements. The advantage of this approach is that the referenced document does not need to be modified when the reference is created (whereas standard Frame cross-references, for example, require modification of both the source and referenced documents).

The encoding used by SoDA has been modified between 2.6.1 and 3.0.2 to remove the previous limitation preventing conversion to and from Maker Interchange Format (MIF), however the approach still has two disadvantages. Objects are renumbered by:

- Hiding and then showing conditional text.
- Cut-and-paste operations (that is, selecting and moving them in the document)

The effect of object renumbering is that existing replicated document elements linked to renumbered objects will be deleted and re-created when the document is regenerated. If you have added information to these elements, that information will be lost.

Add information to elements replicated from the Frame domain with caution. (The templates delivered with SoDA do not call for any information being added to elements replicated from the Frame domain.)

Documents Marked as Changed

If you bring up and then cancel a SoDA dialog, or if you run the **Check Document** command from the **SoDA** menu, your document is marked as having been changed even though it was not.

Books

Duplicate Global Connectors in Dialog Boxes

Global connectors show up twice in dialog boxes brought up in the first document in a book. Select and use the first occurrence of the global connector.

You can avoid this problem by not adding a **SodaBookName** element to the first document in a book. The first document contains the global connectors for the book, so the global connectors will be visible in the first document even though it does not contain a **SodaBookName** element. (You really need

to add a **SodaBookName** element only to other documents in the book that need visibility to the global connectors.)

Full Pathnames in Generated Books

The documents in generated books are listed by their full pathnames, not relative pathnames. If you move a generated book and its documents elsewhere, the book will still be referencing the documents in the original location. (You will also encounter this problem if you have your book and its documents in a Rational Apex subsystem view that is being copied or released.)

Display Issues

The following display issues do not affect SoDA functionality, but they do affect its usability to the extent that they can, unfortunately, be a nuisance.

Flashing Tables

You may notice that documents containing tables flash as the tables are being generated. This flashing does not indicate a problem with the generation.

Scrolling of the Structure View

The structure view is frequently scrolled to the top after SoDA editing operations.

Disappearing Structure View

Prior to document or book generation, SoDA closes the structure view to enhance performance. The structure view may be reopened after generation.