Hoffman-Hays Help Solution (H^{3}S)

Dennis Hays, Adobe Certified Expert/FrameMaker and Adam Hoffman, Oracle Certified Professional

The following is a simple method to implement context-sensitive help from Adobe FrameMaker documents. We realize that there are many methods of accomplishing this using third-part tools, such as ForeHelp and RoboHelp, but there are times when you may not have access to these tools (or don't wish to use them), but desire to provide some type of help system to your users.

This solution allows you to embed context-sensitive descriptors from your documentation into the applications's data base. This method was devised by Adam Hoffman, an expert Oracle™ architect/developer and Dennis Hays, a technical communications specialist and Adobe Certified Expert-FrameMaker. Working together, they were able to conceive, design, and implement this method for the State of New York's Office of Temporary Disability, Child Support Enforcement Redesign project.

Note: While we use Oracle™ as an example, clearly this method could be used by any application development product and the technical personnel should be able to interpolate the instructions below for those application's development tools.

Oracle Preparation

Once the application code has been frozen, the Oracle developer must create a list of item tags representing the blocks, fields, buttons, and other visible items that will need context-sensitive help.

Item Tag Example: CASE_DATA_HOLDER.LEGACY_COUNTY_NAME

These tags are usually constructed so that the information before the period (.) is the block name and the data after the period is the field or button name. Using this construction, it's quite easy to differentiate the blocks from the individual fields.

Create the DOCUMENTATION table with the following script:

```
Create table DOCUMENTATION (
DOC_ID        NUMBER NOT NULL,
TAG           VARCHAR2(80),
HELP_TEXT     VARCHAR2(512))
```

FrameMaker Preparation

After the Oracle developer has created a list of item tags, those item tags are used by the writer to create the context-sensitive descriptors in the following manner
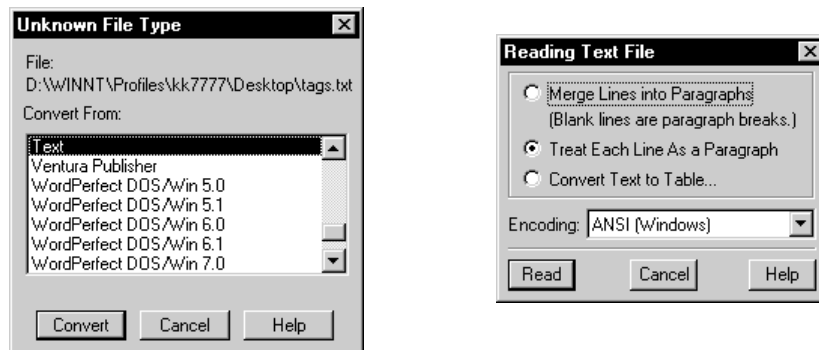

Figure 1: Opening The Oracle Item-Tag Text File

1. Open the item tags received from the Oracle developer in FrameMaker. Usually, these will come in the form of a text file, so open it in FrameMaker with the default: **Treat Each Text Line as a Paragraph.**
2. Save this file as **TAGS.FM**.

Creating a Custom Marker

1. Open the FrameMaker documentation file. This should be a finished (or very nearly finished) file containing the text that will be used as descriptors.
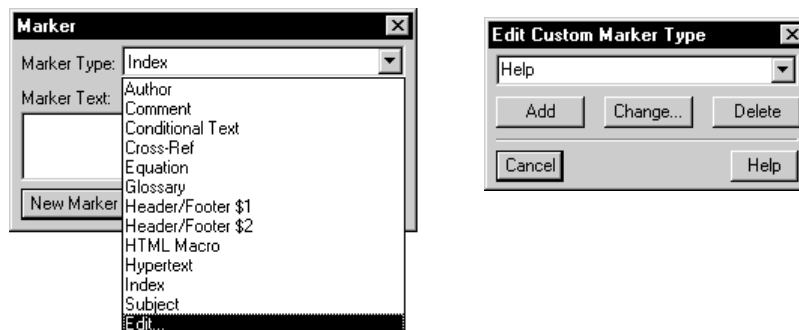

Figure 2: Creating a Custom Marker for Help Text

2. In the Document file, open the **Special** menu and select **Marker...** .
3. Open the drop-down list box at the top and select **Edit...** at the bottom of the list.
4. Type **Help** to add the marker-type "Help" to the list and click the **[Add]** button and then the **[Done]** button.
5. Save the file.

Marking the Descriptor Text in FrameMaker

1. Switch to **TAGS.FM**.
2. Select the first item in the list and copy it to the clip-board (WIN: **Ctrl-c**).

3. Switch to the document file.

4. Open the marker list (**Special>Marker...**) if it is not open.

5. Find the text in your document that describes the item tag you copied in step 2, immediately above.
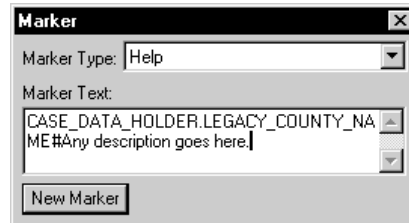
6. Select the descriptor text.

Figure 3: Marking the Text

Note: Make certain the amount of text you select plus the item tag is less than 200 characters. Although FrameMaker has a 255 character limit for the marker text, Oracle (and maybe any other application development system) has a 200 character limit.

7. Place the insertion point at the front of the descriptor text in the **Marker** dialog box and paste the item tag (WIN: **Ctrl-v**).

8. Without moving the insertion point, type a #-sign to separate the item tag from the descriptor text.

Item Tag Example: CASE_DATA_HOLDER.LEGACY_COUNTY_NAME#And this is the pop-up, context-sensitive Help text. pg. 2-4

9. Make certain the **Marker Type** is **Help.**

10. Click the **[New Marker]** button.

11. Repeat steps 8 through 15, but using the next item tag down in the list.

12. Save the document at frequent intervals (safety factor).

FrameMaker Help Generation

Once you have embedded all of the item tag/descriptor text for the context sensitive help, generate this data in a list for two purposes. The first is to create a table from the list for QA (have someone go through this tabled list to ensure the item tags are correctly matched with the proper descriptions). The second purpose is to create a text file for the Oracle developer to use in creating the context-sensitive help.

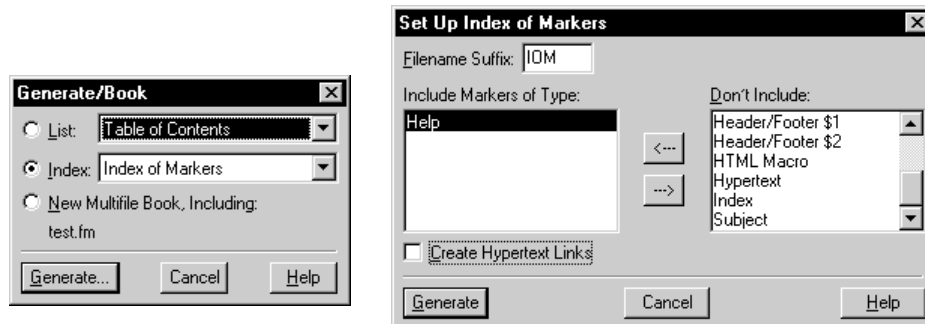1. With your document open, select **File>Generate**.

Figure 4: Generating the Index of Help Markers

2. Click the **Index** radio button and select **Index of Markers** from the drop-down list to the right.
3. Move the "Help" item from the right list in the **Set Up Index of Markers** dialog box (**Don't Include**) to the left (**Include Markers of Type**) by either double-clicking on the "Help" item or by selecting it and clicking the left-pointing arrow
4. Remove the Hypertext check from the checkbox (Hypertext is not necessary).
5. Click the **[Generate]** button.
6. In the resulting **Index of Markers** list, remove the Group Titles (A, B, C, etc.)
7. Save the **IOM** file. (Use any filename)
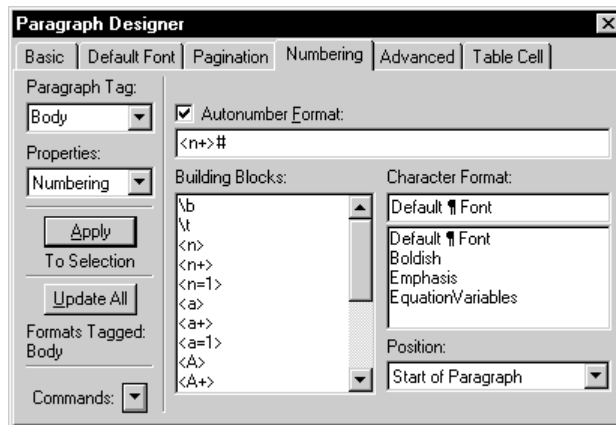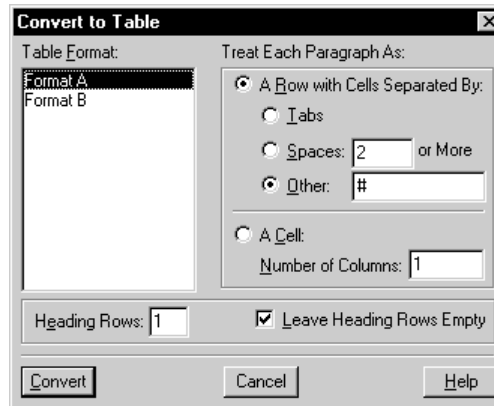8. Open the **FrameMaker Paragraph** Designer (WIN: **Ctrl-m**).

Figure 5: Creating Unique Key Numbers for Oracle

9. Select the **Numbering** properties.
10. Create the autonumber format: **<n+>#**
11. Click the **Update All** button. This adds a unique number to the beginning of each piece of help text, which Oracle uses as a key field internally.

FrameMaker File Preparation

1. While still in the IOM file, select **File>Save As**.
2. Open the **Save As Type** drop-down list box and select **Text Only... (*.*)**

3. Change the file extension, if necessary,in the **File name** text box to have a **.txt** extension.
4. Save the text file.
5. Select the full text in the **IOM** file (WIN: **Ctrl-A**).
6. Select **Table>Convert toTable…**.

7. Select a Table Format.
8. Click the **A Row with Cells Separated By:** radio button and the **Other** radio button under it.
9. Enter **#** in the text box to the right of **Other**.
10. Enter **1** in the Heading Row text box and enable the **Leave Heading Row Empty** check box.
11. Click the **[Convert]** button.
12. Enter Headings in the Heading Row (example: **Item Tags** & **Descriptions**).
13. Print the file and pass to others for QA.
14. When finished QA, make any corrections in your document text and regenerate.
15. Strip the Group Titles (A, B, C, etc.) again (leave the page numbers).
16. Save as a text file and pass to the Oracle developer for incorporation.

Oracle Context-Sensitive Help Incorporation

Receive the data file from the technical writer delimited by something, in our case the '#' symbol. Use SQL*LOADER to insert the data into the table. The control file is:

```
LOAD DATA
INFILE <place your data file name here>
REPLACE
INTO TABLE DOCUMENTATION
FIELDS TERMINATED BY "#"
(DOC_ID,
 TAG,
 HELP_TEXT)
```

Since you will want to control the data from the documentation source, this routine should be used every time an update file is received with the data in the table replaced by the newer data as opposed to updates within the table.

In the Developer application, the following needs to be done:

1. Create an alert (we called ours HELP_DISPLAY), the style should be "Note" and have only one button (BUTTON1) with a label of OK.

2. Create a POP-UP menu with two items (Item level help and Block level help)

3. Create a program unit called GET_HELP, the code is:

```
PROCEDURE GET_HELP (IN_TYPE IN VARCHAR2) IS

hold_it  VARCHAR2(500);
hold_var VARCHAR2(100);
alert_button  NUMBER;

BEGIN

IF IN_TYPE = 'I' THEN
  hold_var := :SYSTEM.CURSOR_ITEM;
ELSIF IN_TYPE = 'B' THEN
  hold_var := :SYSTEM.CURSOR_BLOCK;
END IF;

SELECT help_text INTO hold_it FROM documentation
WHERE tag = hold_var;

SET_ALERT_PROPERTY('HELP_DISPLAY',ALERT_MESSAGE_TEXT,hold_it);
ALERT_BUTTON := SHOW_ALERT('HELP_DISPLAY');

EXCEPTION
WHEN NO_DATA_FOUND THEN
hold_it := 'There is currently no help available about this item.';

SET_ALERT_PROPERTY('HELP_DISPLAY',ALERT_MESSAGE_TEXT,hold_it);
ALERT_BUTTON := SHOW_ALERT('HELP_DISPLAY');

END;
```

4. These items should have the following PL/SQL code attached to them:
   - Item level:
     ```
     GET_HELP('I');
     ```

- Block Level:

    GET_HELP('B');

5. The POP-UP menu should be assigned via the item level property "Pop-up menu" to all items where help is desired.
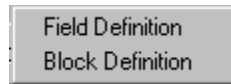

Figure 6: Right-Click Context Menu

When the application runs, the user left-clicks into an item, to gain focus on that item, and then right-clicks the mouse button for the pop-up menu to appear. Then, they can select either **Field Definition** or **Block Definition** and have the help text display as an informative alert.

Note: In our example from above:

CASE_DATA_HOLDER.LEGACY_COUNTY_NAME#And this is the pop-up, context-sensitive Help text. pg. 2-4

we include the page number generated from FrameMaker to act as a cross-reference to the actual hard-copy manual pages.

Updates

From time to time, as you edit the FrameMaker document, generate the list as per "FrameMaker Help Generation" on page3, and then have the developer re-incorporate it as per " Oracle Context-Sensitive Help Incorporation" on page5. This procedure takes only seconds.

Note: If you have any questions regarding this procedure, please feel free to contact:

Dennis Hays (FrameMaker) +1 518 473-7362, email: kk7777@dfa.state.ny.us or dhays@haysdesign.com
Adam Hoffman (Oracle) +1 518 486-5855, email an7490@dfa.state.ny.us or ahoffman@gateway.net