Creating HP KB articles

- About HP KB articles
- Required tools and resources
- HP portals used during KB production
- Setting up structured FrameMaker to use the KB template
- Updating the structure of articles to the current template
- Creating KB PDF and XHTML output files
- Writing process
  - Daily tasks
  - Updating the content of a KB article per a CSO bug
  - Creating a new article
  - Branching an article based on webOS version/device
  - Decommissioning an article
  - Creating sub-tasks
- Graphic guidelines
- Notes on specific KB structure elements
  - Graphic
  - alt_text
  - Procedure
  - Graphic > lightboxthumb
  - Superscript and trademark
- Using the INCLUDES doc
- Using cross-references in KB articles
  - How to create links
  - How to update old-style links
  - What if I am in an agent article and need to use references to both Support and PCS?
- KB writing style
  - KB style variance from UA
  - Title style
  - Voice and audience
  - Writing troubleshooting articles
  - Writing KB how-to articles
  - Writing/editing Policy docs
  - Related topics

- Relationship between writer and SME/reviewer

About HP KB articles

Knowledgebase articles (KBs) present customer support information on HP webOS smartphone and tablet devices. KBs present two main types of information: troubleshooting and how-to.

There are two distinct audiences for KB articles: customers and support agents. Customer articles are published on the Support site (kb.hpwebos.com), and agent articles are published on the PalmCare Solutions site, also known as Solutions and PCS, which is accessible by call center agents, who are part of the Customer Support Organization (CSO). KB articles typically contain both agent and customer information, but may contain just one or the other.

- Support content includes many basic feature/function KB articles. Most of these how-to articles are derived from webOS user guide content. Additional how-to and troubleshooting articles are generated to meet user need in specialized areas (e.g., email setup issues that are too specific and/or complex for a user guide). Support articles contain "KB" in the final HTML file name; for example, article_000_KB.html.
- PCS articles contain troubleshooting information for agents to use while providing support to customers. PCS articles are denoted in the final file name with AT (article_000_AT.html). An agent article may link to relevant customer articles, but to speed access, the link points to a copy of the article loaded into PCS, not to the version on the Support site. PCS copies of Support site articles have _pcs in the file name (article_123_KB_pcs.html).

Required tools and resources

KBs are written in FrameMaker and reviewed as PDFs. The final deliverable for KB articles is XML-compliant HTML files, usually two: AT and KB, or KB and KB_pcs. The FrameMaker files are interim production files. Both the AT and KB final output files are produced from a single FrameMaker source file.

KB article development requires the following tools and resources:

- FrameMaker 9, using structured UI. Additional FrameMaker setup is required for creating XML and XHTML files. See Setting up structured FrameMaker to use the KB template.
    - KB template: The 00_KB_template.fm file contains the formatting, layout, structure, elements, and attributes required writing and producing KB articles.
    - INCLUDES doc: The INCLUDES.fm file contains boilerplate text for agent and customer articles (see Using the INCLUDES doc).
- Acrobat Pro: Required to edit PDFs and to make PDFs reviewable with Adobe Reader.
- Microsoft Visio to edit diagrams in policy KBs.
- Screen capture software (any)

- Graphics SW for cropping screenshots, making minor edits, etc. (any semi-current version will work).
- Fonts: Download and install all of the fonts from \\Reference_Tools\Fonts.
- UA Style Guide
- Creating HP KB articles (this document)
- Attribution block and General guidelines
- Palm webOS Glossary (status?)

HP portals used during KB production

Upon your start at HP, work with your UALS project manager to get access to these internal portals:

- SVN
- Jira
- PalmCare Solutions (aka PCS or Solutions)
- Sharepoint
- CSO Wiki

SVN

SVN is a version control software program. TortoiseSVN is the required application. You can download TortoiseSVN from **http://tortoisesvn.net/downloads.** Check with your UALS project manager for the supported version of TortoiseSVN.

After you install TortoiseSVN, request access from your UALS project manager to these folders (and subfolders and files of http://subversion.palm.com/public/UALS:

- **http://subversion.palm.com/public/UALS/Nova**

  This folder contains the source, in-progress, and final, for most UA deliverables. Of interest to KB writers are the various user guides and /Castle/KBArticles.

- **http://subversion.palm.com/public/UALS/Reference_Tools**

  This folder contains UALS department resources such as fonts, templates, style guide, and glossary.

Jira

Jira is the issue tracking system. Your UA project manager assigns KB projects based on Jira issues. Your UA project manager requests your Jira account. To use Jira, just type *jira* in the address line, or point your Web browser to **https://jira.palm.com/**.

Jira issues are handled in order of Impact (Critical, Moderate, Low).

You can create a filter in Jira to bring up your assigned issues.

PalmCare Solutions (aka PCS or Solutions)

PalmCare Solutions is an internal-facing knowledgebase portal for the Customer Support Organization (CSO).

Your UA project manager requests your PCS account. You can use PCS by pointing your Web browser to **http://solutions.palm.com/**.

On the PCS home page, you can use the **Solutions Search** field to search PCS for articles by article number. You can use the **palm.com Search** field to search the Support site for customer articles by one or more keywords. Also, you can scroll to the bottom of a PCS article to see the link to the customer article.

At the bottom of a PCS article, an agent can enter comments about the feature or issue. When updating an article, check the corresponding PCS article comments to see if the agents have requested other changes to the article or if they've provided additional information about the article you're working on.

Sharepoint

UALS maintains a Microsoft Sharepoint site with project information and other helpful resources. You can use the UALS Sharepoint site by typing *docs* in the address line, or point your browser to **http://infocenter/departments/marketing/docs/default.aspx**.

CSO Wiki

The Customer Support Organization wiki site contains information on webOS updates. Contact Corey Marugg for access. You can use the CSO wiki by pointing your browser to **https://wiki.palm.com/display/CSO/Home**.

Setting up structured FrameMaker to use the KB template

IMPORTANT For historical reference, the sections below provide the paths in Reference Tools to the various files used to set up FrameMaker. These files will be provided to you.

Close FrameMaker before updating it.

You need to update two Frame files and add a folder containing the new dtd file. You can find the Frame files at the top level of the 00_KB_Template folder. Proceed as follows.

00_KB_template.fm

The template file is 00_KB_template.fm. The "00" makes the file float to the top of the Source directory for easy access. The first time you use it, you may have to update the graphics path. You can rename this file and put it anywhere you want. Your local Source folder is a convenient place, but don't SVN control this file.

In SVN it is located at:

http://subversion.palm.com/public/UALS/Reference_Tools/FM8_Templates/KB_Templates/External_processing_templates/

structapps.fm

In SVN this file is located at:

http://subversion.palm.com/public/UALS/Reference_Tools/FM8_Templates/XML_output/structapps.fm

This file needs to be placed in a specific location in your local FrameMaker directories:

C:\Program Files\Adobe\FrameMaker9\Structure

"external_kb_agent" folder

The folder external_kb_agent contains the dtd files external_kb_agent.dtd and rules.txt. Put the folder here:

C:\Program Files\Adobe\FrameMaker9\Structure\xml\Palm

NOTE If you find no Palm folder, create it.

Setting up the external XHTML processor

NOTE The detailed instructions for setting up a computer to create XHTML files were originally provided in XHTML_XSL_transformation.pdf (1_UALS_Subversion\Reference Tools\FM8_Templates\KB_XSL_Templates). This documentation was provided by Scriptorium, the company that created the KB template and associated files for processing XHTML. The procedures are now provided in this section, with additional notes added.

Installing Xalan and J2RE

To transform XML to simple HTML with the XSL templates, you need an XSL processor called Xalan.

To install Xalan, follow these steps:

1. Download Xalan from apache.org:
   http://xml.apache.org/xalan-j/
   Download xalan-j_2_7_1-bin-2jars.zip from the available files.
2. Extract the folder "xalan-j_2_7_1" to the C: drive.

IMPORTANT This documentation assumes that you installed Xalan version 2_7_1 in the root of your C: drive. If you have installed a different version or saved it in a different location, adjust the directory paths in set_paths.bat to point to the correct files. See Setting up the batch files for more information about this file.

Xalan needs Java as a foundation. We recommend that you use the Java 2 Runtime Environment, Standard Edition 1.6 (JRE).

NOTE Most things done at HP involving devices require you to have whatever the current version of Java is, which works fine with Xalan.

If you don't have Java already installed, follow these steps:

1. Download the JRE from:
   http://www.java.com/en/download/manual.jsp
2. Run the executable file to install the JRE.

Setting up the XSL templates folder

IMPORTANT The instructions in this section and the next two sections—Setting up the projects folder and Setting up batch files—are done for you. The setup files provided include a kb_agent folder fully stocked and ready to go. Just drop it on your C: drive. This is the folder to save your XML files to. You don't need to do the following procedures.

The XSL transformation process uses relative paths between the input files, XSL templates, and the output folder. To set up the correct directory structure for use with the XSL templates, follow these steps:

1. Create a new folder on your C: drive. Name this folder "kb_agent."
2. Create a new folder within the kb_agent folder. Name it "xslt."
3. Copy the .xsl and .xml files from your local version of the KB_XSL_Templates folder to the xslt folder within the kb_agent folder on your C: drive.
4. Make sure you have the following files:
   - char.xsl
   - filename.xsl
   - graphics.xsl

- html_hf.xsl
- hypertext.xsl
- pages.xsl
- para.xsl
- simple_tables.xsl
- start.xsl
- strings.xsl
- variables.xsl
- xrefs.xsl
- loc_strings.xml

Setting up the projects folder

Each article you convert requires two folders to hold the XHTML output. To create the output folders, follow these steps:

1. In the C:\kb_agent folder, create two new folders: "articles" and "articles_noatt."
2. Copy the XML files you saved from FrameMaker into the C:\kb_agent folder.

Setting up the batch files

The XSL transformation is run using batch files. To set up the batch files, follow these steps:

1. Open your local SVN version of the FM8_Templates\KB_XSL_Templates folder.
2. Copy the following batch files from the Batch folder to C:\kb_agent.
   - set_paths.bat
   - setup_computer.bat
   - KB.bat
   - AT.bat
   - output_noatt.bat

Updating the structure of articles to the current template

The KB template went through a major revision after it was created. Attributes were changed and many new ones were added. If you are working on an article that has not been updated since the release of webOS in the fall of 2009, there are a number of things to update to make the document compliant with the current generation template. Articles that have not been updated to the current template have the Siebel codes list at the beginning. Articles on the current template have it at the end.

There have been a number of smaller ongoing updates to the current template, the most recent in April 2011. Unless you know an article is current, do a quick update by doing steps 1 to 4.

NOTE To create a new article with the new template, do only steps 5-9, 11, and 15.

1. Open the new template file and the KB article file. Make the KB article the active window.
2. Select **File** > **Import** > **Formats**. Select the new template file, and select all the options. Check both boxes for removing overrides.
3. Select **File** > **Import** > **Elements**. Select the new template file, and select all the options. Check both boxes for removing overrides.
4. Select **Format** > **Page Layout** > **Apply Master Pages** and reapply all the master pages. You should see new headers filled with attribute entries.

NOTE Attributes that existed previously are not mentioned below. Enter/select them as before.

5. In the Structure View window, enter a value for **device_os_version** (1.x,2.x, 3.x,). Note the following:
   - If an article is generic; that is, its content applies to all webOS devices and versions, enter the webOS release as 1.x, 2.x, 3.x. Otherwise the specific version to which it applies.
   - If the change requested is related to a future software update or carrier release (e.g., 2.2.0), update the OS version attribute to the release (meaning *this article is current with 2.x*) and state in the publishing sub-task that the article should not be released until the relevant software version is released.
6. Select a value for **product1**, **product2**, etc. representing all the products to which the KB applies. See header above for how these are used. Historically, nearly all KBs applied to all current devices, with just a few KBs applying to only single devices. Beginning with the release of the TouchPad in June 2011, agent and customer KBs branched according to webOS version and device (smartphones and tablets). Policy KBs will typically still apply to all devices and webOS versions, but some are device specific (such as how repairs are handled).
7. Enter a value(s) for **carrier** from the following approved values: HP, Sprint, BellMo, SFR, Telefonica, TelCel, Verizon, ATT, SFR, Rogers, Vodafone. Most articles we write are generic and apply to all of the carriers, in which case you would list the current ones, separated by commas. The template file should have all of the values already populated. Just check to see if you need to add or delete to make current.
8. Select a value for **category** from the list.
   - For existing articles, the category can be found in PCS at the end of the article in a section called **Tag page**. You'll see a line that gives the article number, category, type, and web OS version (e.g., article_105, Email, How_to, Pre, webOS_2.1.1). Note that just above the Tag page section is the customer KB article ID, as a link (most of the time).
   - For new articles, the category may be listed in the bug. If not, choose the most appropriate one and call attention to it during the review cycle if you have a question.
9. Old, never-updated articles may have a **product** attribute. Delete it. Select it, click the Delete button by INVALID ATTRIBUTE, and accept the default.
10. For both the Agent and KB structured sections:

a Delete the preexisting value for **article_type** (**agent** and usually **breakfix**).

b Select one of the four new values: **System**, **Policy**, **Troubleshooting**, **How_to**.

NOTE Policy docs are articles that define CSO support policies. System articles are a type of policy article that documents procedures such as how agents use the Siebel system to log support calls.

11 Expand the Agent section and expand the **AgentIntro** element if necessary. Cut the **SiebelCodes** element from it and paste it at the end of the file, after the KB section.

NOTE If you are updating an article that has only the KB template section, you need to copy the SiebelCode element from another article that has an Agent section.

12 Delete the **AgentIntro** element. Your document should now be valid (no red).

13 In the content view, delete the three empty paragraphs at the beginning.

14 Save the file and check that the header information has updated.

Creating KB PDF and XHTML output files

This section describes the structured KB FrameMaker template and the process for creating PDF review files and final XHTML files (XML compliant HTML files).

- Creating PDF files
- Creating XML files
- Creating XHMTL files
- Updating XHTML output

Creating PDF files

While working on an article, save in-process/review PDFs locally. (You can use the WIP folder if you want to back them up.) When you're finished with an article, clear any change bars and save a PDF to the Final_PDF folder in SVN (the root folder, of the 1x, 2x, 3x folder as appropriate). Most reviewers have Adobe Acrobat now, but not all. It's good to save the review copies so they can be edited with Adobe Reader (**Comments** > **Enable for Comments and Analysis in Adobe Reader**).

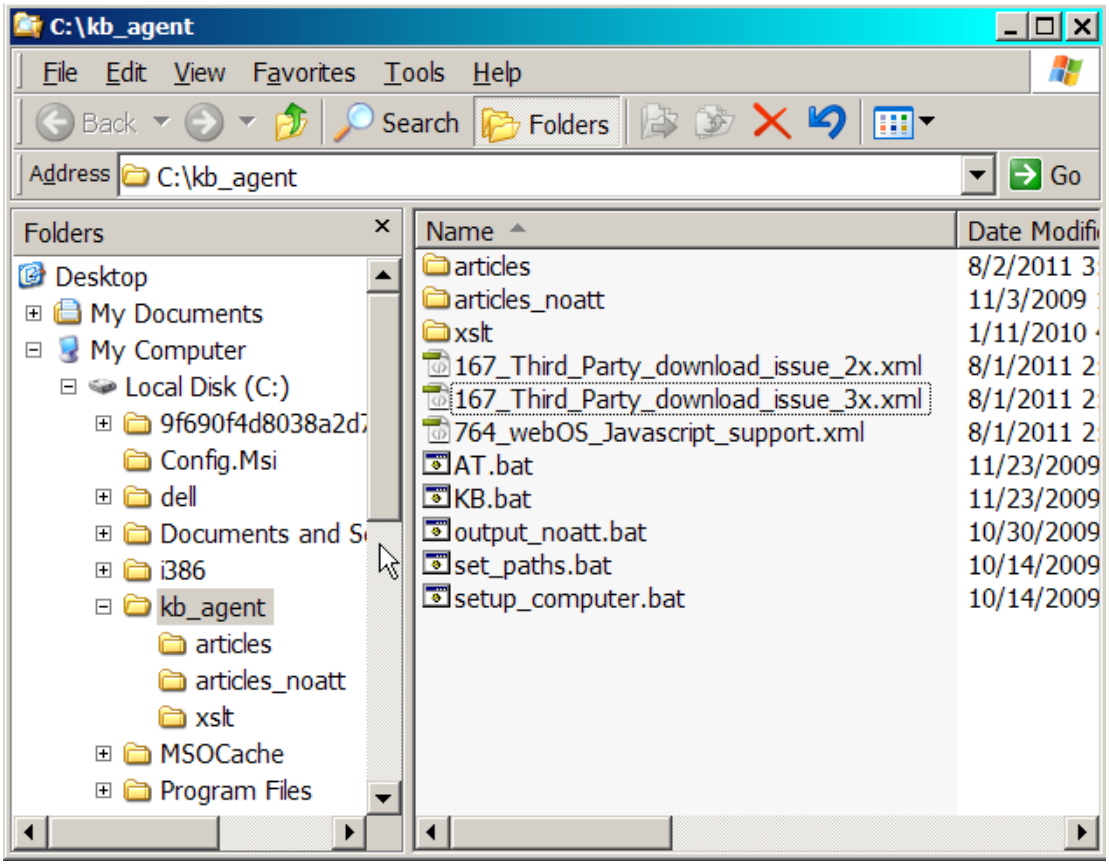
Creating XML files

When you're finished with an article and ready to publish it, save it as an XML file.

Under **StructuredTools** > **Set Structured Application**, check that **ExternalKBAgent** is selected (which so far seems to be always selected by default). If you don't work on other projects using other templates, this option will stay your default.

Select **File** > **Save As** > **XML**. Save the XML files to **C:\kb_agent**, the directory created when you set up FrameMaker for KB production. The kb_agent folder is self contained and can be put anywhere, but this document assumes it's at the C: level.

NOTE It's a good idea to use **Element** > **Validate** on new articles to check the structure. Existing KBs have already been validated and will be fine unless some new input causes trouble, which is not uncommon. If it does, Save As XML will fail and you'll have to fix the issue. The structure view always shows structure problems in red.



Creating XHMTL files

1. In the kb_agent folder, double-click **setup_computer.bat**. A command window opens.
2. If the content being transformed contains both an Agent and a KB article and the product attributes are to be included as HTML comments within the output, type **at** > **Enter** in the command window. If the content being transformed contains only a KB article and the product attributes are to be included as HTML comments within the output, enter **kb**. If you are creating output without the product attributes included, enter **output_noatt.bat** (we've never had to do this).

NOTE As long as you keep the .bat suffix, you can rename these BAT files.

3. You are prompted to type in the name of the file to convert. Enter the KB number and press tab; the rest of the file name is added. Backspace to delete the .xml extension, including the period.

4. Press **Enter** to convert the .xml file into XHTML output. The process is quick. You'll see a processing line followed by a return to the C:\kb_agent prompt.

   The produced HTML files are placed in the articles directory inside the kb_agent folder.

5. Navigate to the folder containing your output. Verify that separate files for the Knowledgebase and Agent content were created. You may want to put shortcuts here to the folders that receive the final files. Then you can just drag them there after they are created.

NOTE While in the command window, you can press the up and down keys to cycle through previously typed commands. If you have already typed in the name for the XML file, you can use this feature to avoid typing it again.Press the up or down key until you see the name of the XML file on the command line, then press Enter.

Updating XHTML output

If you need to update the output because of changes made to the FrameMaker file, you need to save the FrameMaker file to XML again and rerun the entire XSL process. Existing XHTML files in the articles and articles_noatt directories are overwritten by new files without a warning message.

Occasionally the processing produces simple errors such as extra line breaks in HTML that have to be corrected manually. This typically happens when the structure gets more complex than it was designed for, such as multiple para elements between list items. The output script has been modified to allow more flexibility, so errors are infrequent now.

In general, though, the rule of thumb is not to make changes in the HTML. Fix the issue in FrameMaker. This may involve rewriting to simplify/change the structure.

Writing process

The tasks that follow cover creating and updating KB articles.

> TIP Check to see if you can update an existing KB article before creating a new one. Also look for articles related to one you are creating/updating that may also need to be updated.

Daily tasks

1. Check for critical bugs in the Support Content queue in Jira and/or use the relevant Jira filters and custom tag **new_writer**. Either assign it to yourself or confer with other writers as necessary. Yes, this a bit vague, but with only one writer, and then two, simple communication and doing the job worked. Now that there are multiple writers again, the process will become more defined. (08/2011)
2. Update your files in SVN to make sure you have the most current files. If you don't, well, SVN can be harsh.
3. At the end of the day (or during the day if you fear an outage, have written that really great paragraph you don't want to lose, etc.), commit any changes to SVN. Articles do

not have to be complete to be committed. SVN functions as an in-process backup as well as source control.

Updating the content of a KB article per a CSO bug

Bugs against KBs may be assigned to you, or you may assign them to yourself directly from the Support Content queue in Jira.

To update an existing article:

1. Open the bug assigned in Jira. Click **Assign To Me**.
2. Plow through the Jira UI to get the bug to the status "implementation." Click the **Analyze** button > **Analyzed** button > **Implement** button > **Implement** button. The Implement button changes to Implemented, which you'll click when you're finished. The bug now shows up as being worked on when other people check its status. Check the Assignee under People at the right of the screen to make sure the bug is still assigned to you. Depending on what order things are done in Jira, the bug may self assign to someone else.
3. Open the bug and review it for the following:
   - Scope of change required. You may need to open and read the KB article to do this, or you may be able to tell just from the input in the bug. Has enough information been provided? If not, add a Comment and ask for the missing input. If you don't receive a response, send an email directly to the SME. If you still don't get a response, assign the bug directly to the SME.
   - Check for links to related bugs. Read those bugs and address any items that affect the articles) you are updating. Assign those bugs to yourself. Confer with the other writer if the bug is already assigned. There may be related bugs that have not been linked. To check for these, click the article_XXX custom tag, which will show a list of all bugs for that article. ("Custom tags" is nine items down on the left under Details. The entries are links.)
   - Click the article number custom tag to show all other bugs concerning the article. It's not uncommon for bug reporters to file bugs against a KB without checking to see if there are other bugs against it already.
   - For existing PCS articles, check the article in PCS to see if there are relevant comments from agents.
4. Update your SVN Source files.
5. Open the KB_template.fm and INCLUDES.fm files.
6. If necessary, update the article to the current template (see Updating the structure of articles to the current template for the details of this process.
7. Compare the conventions used in the article to the UA style guide, this document, and to the INCLUDES file, and update as necessary.
   - Some existing articles may use Sprint style, which is incorrect for KBs, and some standards have changed since articles were created.
   - Some of the early articles (002–300 approx) may have stylistic errors in terms of the capitalization and bolding of the names of buttons, menus, etc. In some cases the

standard has changed, and in others the names for buttons. (Some early articles were not written by tech writers and did not go through editing.)

8 Replace any placeholder links with real ones. Includes initially used placeholder links (e.g., **How to - 340**) Change these to the correct style (see Using cross-references in KB articles).

NOTE Note: At this point (08/2011), these links are left only in articles that have never been updated, which usually means they have never been used, which means they are probably not needed, or if they are, substantial editing and rewriting will be required to bring them up to code.

9 Use change bars to track substantive content changes, not every change (e.g., turn off for spell check). If there were a few important changes to content, but the whole article has change bars, the change bars don't help the reviewer.

10 Add the required content. During this process you can use email to communicate with the document requester/reviewer, and you can also use Comments in Jira, whatever in your judgment is most appropriate. Significant steps along the way should be documented in Jira. Copy text from email and paste into a Jira comment if necessary.

11 In the Jira bug, create an EDIT sub-task and attach a PDF of the file (see Creating sub-tasks). Add the custom tag **edit**. Assign the EDIT task to **scott.eldredge**. Change the status of the sub-task to Implement. Include comments to the editor if you have any. Conduct all communication relating to editing via clearly stated Jira comments and attachments.

12 After receiving and incorporating editor feedback, in the Jira bug, create a REVIEW sub-task for the reviewer and attach the revised PDF (see Creating sub-tasks). Add the custom tag **review**. Change the status of the sub-task to Implement. If necessary, annotate the PDF for the reviewer with any comments that will ensure that the key information gets reviewed properly—ask questions, explain changes, etc. If you are substantially rewriting someone's input, it can be helpful to explain why you did what you did.

NOTE Because articles may go through several review cycles, it may be useful to add the date, your initials, rev1, etc. to the PDF file name. Reviewers may return the PDF with their initials added, or they may not. Adopt your own process here.

13 When the article is ready for publishing, create the final PDF (without change bars) and XHTML files (see Creating KB PDF and XHTML output files). Commit the source files in SVN (including new graphics), PDF (don't forget watermark), and final XHTML files, and create the required publishing sub-tasks. Place files as follows. Use the root, 1x, 2x, 3x folders as needed.

- New icons and screenshots > Final_Graphics
- Final_PDF > [000_FM file name].pdf
- Final_XHTML > Final_AT > article_000_AT.html
- Final_XHTML > Final_KB > article_000_KB.html
- Final_XHTML > Final_KB_pcs > article_000_KB_pcs.html

- Source > 000_[FM file name].fm

14 After the PUBLISH and LOCALIZATION sub-tasks have been created, click Implemented on the parent bug. You're done.

Creating a new article

To create a new article:

- Start with the template file or another article file updated to the KB_template.fm and do a Save As. You can also start with another article that is similar in some way; be sure to update and change all the attributes in the Structure View window to those of the new article.
- If you are branching an existing article, do a Save As in FM, commit the new file, and SVN rename the existing file as necessary. See Branching an article based on webOS version/device.
- If either the Agent or KB part of the template is not used, delete it.
- Update the attributes as necessary.

For detailed information about setting attributes, see Updating the structure of articles to the current template.

Branching an article based on webOS version/device

NOTE Precise instructions about what needs to be done to branch a KB will be provided in the bug.

KB articles were originally planned to be generic. An article would be applicable to all webOS devices and webOS versions. Articles were written with a more generic style, and as the UI and functionality were changed from one release to the next, articles were revised to suit, but kept generic. A UI-specific step might be rewritten to be task oriented to avoid mentioning the UI, as in the following progression:

Open Launcher > Device Info > application menu > Tests > Interactive Tests > Power

Open Launcher > Device Info > application menu > Diagnostics > Interactive > Power

Open Device Info and run the interactive test for power.

Another tactic would be to create steps with conditional actions or substeps to account for different UI or device options:

1 If your device has a removable battery, remove and reinsert the battery (see Replacing the battery on a Pre phone [44919, 303] or Replacing the battery on a Pixi phone [24398, 526]). The HP Veer does not have a removable battery.

So, what does this mean to us today, especially if we're not KB historians?

This process worked for about two years. When webOS 2.0 came out, it did not apply to all devices, as all previous versions had. Early Pre and Pixi devices were left behind at webOS 1.4.5. UI and feature changes in the 2.x line made it necessary to break out some articles into 1x and 2x versions, or early and later 2.x versions. With the Touchpad, the

only device with webOS 3.x, and all the hardware differences associated with being a tablet, we decided we had to create a separate branch of KBs just for the TouchPad and subsequent 3.x tablets.

This is the process we are in now as we continue to create the first full set of TouchPad KBs.

When branching a single KB into multiple KBs, the 3-digit KB number and article title are kept the same. The PCS and Support sites can load, maintain, and track different versions of an article and display the correct one based on the device a user selects when searching in Support, or an agent searches in PCS. But to do this, we have to keep the KB number and title the same.

The Frame file is duplicated to create 1x, 2x, and 3x versions by adding _1x, _2x, _3x to the end of the file name. The 1x version is generally just the 2x content; 1x content is not typically created until needed.

Decommissioning an article

Articles can become obsolete for a number of reasons. For example, a feature may be discontinued, or a troubleshooting article may have been written to provide a temporary solution to a problem that was resolved a few hours later with a webOS update.

Follow these steps to decommission an article:

1. Assign the bug to yourself (if you are not the creator).
2. Salvage content that is still valid for use in other articles.
3. Make sure all links TO the retiring article from other articles are removed or replaced with relevant links. Follow the standard procedures for working with and publishing articles. To find links:
   - **a** Follow the links in the retired article and check those articles for reciprocal links. Also check those articles for links to other articles that might link to the retired article.
   - **b** Use Windows Search in the Final PDF folder to look for instances of the KB number for the retired article.
4. Move the FrameMaker source and Final PDF in the **Archive** folder (right-click and drag and select **SVN move versioned files here**). Do an SVN Commit.

   http://subversion.palm.com/public/UALS/Nova/Castle/KBArticles/enUS/Archive

5. Add the **kb_retire** custom tag to the bug if necessary.
6. Assign the bug to solutions to complete the next part of the process—removing the files from PCS. Solutions will do its part and assign the bug to Support Knowledgebase, who will do the same and assign the bug to Localization, who will remove the localized versions and then close the bug.

Creating sub-tasks

After you complete an article and have posted the final XHTML and PDF to their folders, you need to create sub-tasks in Jira telling the appropriate people that the article is ready to be reviewed, is posted and available for loading into the Support environment (kb.palm.com), the PCS environment, and Localization (ATs and KBs).

1. In the bug for which the article was created/updated, select **More Actions** > **Create Sub-Task**.
2. Assign the subtasks as follows:
   - **Support Knowledgebase** (customer articles).
   - **solutions** for PCS (all article types).
   - Fabien Desmole for localization (all article types, but only Critical Impact 1).
3. Select **KB Article Public** or **KB Article Agent** for Sub-Task Categories.
4. In Summary, provide the article ID and project the article is tied to, if any. If the bug is Critical and is being published, include this in the Summary. DO NOT enter the bug number on the Summary line. Examples:
   - 1A PUBLISH: TOPAZ 052_3x_AT (for solutions)
   - 1A PUBLISH: TOPAZ 052_3x_KB (for Support Knowledgebase)
   - 1A LOCALIZATION: TOPAZ 052_3x_AT, 052_3x_KB (for Fabien Desmole)
   - 1A EDIT: TOPAZ 052
   - 1A REVIEW: 052_AT, 052_KB
5. In Description, provide the file names and the titles of the articles and indicate the following if it applies:
   - Project information from above again, or expanded (e.g., Do not publish until 3.1 is released on October 31.)
   - If the article is NEW.
   - If the title has changed (for non-version KBs). (Titles are stripped in the HTML and added manually in PCS and Support.)
   - The file names of new graphics.
   - Any other pertinent information that will help, such as the article contains links to other new articles that have not been loaded yet (and so for which you don't know the URL).
6. Add the "localization" custom tag for that task.
7. Click **Create**. In the new bug window that appears, click **Implement**, and then **Implement** again in the next window.

Graphic guidelines

The standard file type for graphics is .png.

Size

Icons and in-line graphics: 18 pixels in height. Import into FM at 100 dpi.

Size refers to the size of the source image, not the dpi at which it is imported into FrameMaker. The FM dpi setting has no affect on the HTML output.

Screenshots: 146 pixels wide. Import into FM at 100 dpi. For smartphone screenshots, do not show the title bar of a screenshot unless the information in the title bar is required for the procedure. The title bar shows carrier/network-specific information that limits the applicability of the image. Crop the title bar from screenshots you create. If you want to save an uncropped version of the screenshot, put in Final_Graphics_FUL

Whole-device images: 186 pixels wide. Import into FM at 150 dpi. (This spec is likely to change if we go to more generic graphics that don't show the whole device.)

Bigger screenshots such as from Windows/Mac applications: 477 pixels wide (this is 3/4 of the KB article space). The widest possible size, which will fill the entire article space: 580 pixels wide. (Some applications, like Siebel, have gigantic screens that are barely legible even at the maximum size, so 477 doesn't work.) A new attribute of the **Graphic** element called **lightboxthumb** was created recently to deal with large screens. Selecting **lightboxthumb** causes a thumbnail version of a large image to be created, which when clicked displays the full size image in a pop-up or another window.

The dpi at which the image is imported into FrameMaker doesn't matter for images that appear onscreen. The screen only cares about the width or height (in pixels) of the final image. You can't correctly size images for web view by changing the dpi in FrameMaker.

Using graphics

Don't overdo screenshots, and don't use just to have pictures. You can use to orient the user to what he/she should be seeing on the phone, but you don't need to illustrate everything. KBs don't need as many screenshots as UGs.

Remember, for KBs, the user has a phone to look at. For ATs, the agent does not have a phone to look at, but does have access to a simulator. You can assume the agent has access, which is not access as quick and easy as an actual device. So if an image is critical to the procedure, provide it. Article 504 is an agent troubleshooting article that has graphics.

Don't use screenshots to document the screens of a wizard that guides the user through a procedure you are documenting. SMEs may provided all these images as input for a bug, but don't use them unless necessary. They are provided to document the process and help the writer, who may not have access to the SW being documented.

You don't have to include screenshots just because they were provided by the person who requested the article. The writer makes the decision about what is needed.

Notes on specific KB structure elements

Graphic

The **Graphic** element has a new attribute value (04/2011) for style called **lightboxthumb**. It is used when over-sized screenshots are used, such as for tablet devices or large

application screens like Siebel or Windows. A thumbnail appears that the user clicks to display the image in another window.

alt_text

The **alt_text** element is required for accessibility reasons, so visually impaired people can have the documentation read by a screen reader. The alt_text assigned to a graphic is what would be read by a screen reader when a graphic appears. The value is usually "null," which means that nothing required or of benefit would be added to the sentence by having alt_text for the graphic read aloud.

For example, "Press **Option** " does not need a readable description of the icon for the sentence to be understood when read, because the word *Option* contains all the meaning. But some icons don't have names. "Tap and tap **Assign to contact**" doesn't make sense when read unless there is an alt_text description for the icon. In this case, the value for the **alt_text** would be something like "the Assign-to-contact icon." When read aloud by a screen reader, the sentence becomes "Tap the Assign-to-contact icon and tap **Assign to contact**,"which isn't great literature, but at least makes sense. The alt_text phrase can usually be paraphrased from the file name of the graphic icon. If it can't, well, you are paid to be a writer.

IMPORTANT Remember that the alt_text is to be read aloud, so you can't just name the icon; you need to include articles like *an* and *the*, whatever makes the whole sentence readable.

Screenshots and large graphics use "null" for alt_text, because there's no way to describe them meaningfully. A flowchart would turn into a convoluted story, a description of Card View would be as useful as you imagine, etc.

When this requirement was added to the template, existing articles were updated to include values (otherwise the structure would not validate). Since graphics are copied from one document to another, values have come along, but occasionally new icons are added, etc., and the alt_text element needs to be filled in correctly.

All HP documentation must meet this accessibility requirement, so in KB-land we follow the same rules.

Procedure

Procedure is an element from the UG world and is the same as **Bold** in KBs, which should be used instead. The use of Procedure does not invalidate a document ad processes properly.

Leadin

The UG template has more elements than are used in KBs. An example is Leadin. While it does not invalidate the structure of a KB, it does not appear in the proper font. It does appear in the Elements catalog because the template was derived from an existing one. If you see a Leadin element in a KB, unwrap it. If you want to create the style of a bold leadin phrase, just manually apply Bold to the desired text.

Graphic > lightboxthumb

When exactly to use the Graphic element attribute lightboxthumb is still being decided. More coming on this exciting new capability. For now, if you come across a case where you need to use a large image that may benefit from a thumbnail/popup, check with Dan Lavelle (Solutions) or Bonnie Lewellyn (Support) to see if they'd like it used in the KB in question.

Superscript and trademark

HP does not use trademark symbols with the same frequency as Palm did. Working with the trademark symbol will mostly involve removing it. However, note that KB titles should not have trademark or superscripts in them even if they might be required. The processing involved in producing XHTML strips them out. If you see any such marks in titles, remove them.

Using the INCLUDES doc

This is the file named INCLUDES.fm, located in Source in SVN.

INCLUDE statements

The INCLUDES document was created to store standard phrases that were used when writing troubleshooting procedures for agents. The text that follows the colon in an INCLUDE statement is can be replaced in a batch process in all AT articles, so if, for example, a change in policy requires that agents change the way they handle support calls for which they cannot provide a solution, all instances of INCLUDE NOT SOLVED can be updated with new text in all PCS articles.

The INCLUDES are used only in troubleshooting flows. Cut them from structure in the INCLUDES doc and paste them into the article yo are writing. Note that you may need to change the structure element in your article from what it is in the INCLUDES doc.

Older articles may have incorrect INCLUDES text, but almost always have incomplete links in the INCLUDE statements. Don't update just the links. Copy the whole element when you need to update an INCLUDE.

DO NOT edit or change INCLUDE statements, in your document and especially in the INCLUDES.fm file. Only Scott and Midyne work in that file. However, the INCLUDE statements do not cover all possibilities that can occur in troubleshooting flows. You can tweak the language of an INCLUDE statement if you must, but then you must remove the INCLUDE part of the statement (the all-caps colon phrase like INCLUDE NOT SOLVED:). If you don't, your rewrite will be replaced with the stock phrasing when the document is loaded.

Old documents may contain the following Includes that are no longer used:

INCLUDE SOLVED SOFTWARE: In general, delete steps that use it. Checking for a software update is no longer a procedural step.

INCLUDE NOT SOLVED SOFTWARE: In general, delete steps that use it. Checking for a software update is no longer a procedural step

INCLUDE DASHBOARD: Delete step. Dashboard is gone.

INCLUDE TSO: Replace with INCLUDE SOLVED

Common phrases

The statements in the "Common phrases" section are used like the INCLUDES. They provide a way for all writers to say the same thing the same way. The phrases included are ones that showed up a lot in troubleshooting articles. Again, cut them in their entirety from the Structure View. They include correct **How to** links.

Using cross-references in KB articles

There are subtypes within the main categories of KB articles. The type of article and its final destination determine how cross-references are handled.

- In agent articles, most links are **How to** or **Policy**. Some are to the title of the article in PCS (commonly in the form "Go to **My battery won't charge**").
- Bold the **Website** element in PCS articles. Do NOT bold it in Support articles.
- Check actual titles in PCS and Support. They sometimes get changed on the back end from what is in the Frame file.

Article types, locations, and cross-referencing

Content type	**Where final files are loaded**	
	PalmCare Solutions (PCS)	**Support (palm.com)**
Troubleshooting (agent and customer)	Q&A troubleshooting flow created in Agent part of template. KB cross-refs are to PCS location of KB, because KB articles are also loaded into PCS. These links are created automatically during HTML file creation and are used in the file with the KB_pcs name.	Created in KB part of template, standard writing style. Cross-refs are to Support site location or Internet. Contain no cross-refs to PCS.
How to (customer)	Created in KB part of the template, standard procedure style. Two versions are output: the KB file is loaded into Support, and the KB_pcs file is loaded into PCS. KB cross-refs are created automatically to both the Support location and the PCS location of other KB articles.	Created KB part of the template, standard procedure style. Loaded into Support. KB cross-refs in output are to Support location.

Article types, locations, and cross-referencing

Content type	Where final files are loaded	
	PalmCare Solutions (PCS)	**Support (palm.com)**
How to (agent)	Created in Agent part of template. Not as common as troubleshooting flows. Standard writing style. KB cross-refs are to PCS location of KB.	Not loaded.
Policy, messaging for agents	Created in Agent part of template. Standard writing style.Cross-refs are only to PCS location. Not likely to contain KB cross-refs, but would be to PCS location.	Not loaded.

How to create links

1. Create links to the Internet the same way as in UG, using **Website** and **Hypertext**. If the Website element contains the Hypertext element, it is processed as a standard URL.
   - **a** Select the phrase that is to be the link and apply Website.
   - **b** Click within the link and apply Hypertext.
   - **c** In the Hypertext dialog, enter message URL and put the URL after it. Look at this link for an example: Google.
   - **d** If the link is in an AT article for use by agents (including policy KBs), apply the Bold element to Website.
2. For links in AT articles, select the link text and apply **Website**. Select a value for **PCS_location** from the list, and enter our KB number for the PCS_ID. Typically the PCS_location will be **webOS**. Policy and System correspond to the same two attributes for article_type). Bold the AT links as before.
3. For links in KB articles, select the link text and apply **Website**. Enter the 3-digit article ID (our KB number) for **PCS_ID** and the 5-digit kb.palm.com article ID for **KB_ID**. Also enter the PCS_location. The article ID numbers are made visible for review purposes. They are removed from the HTML automatically.

NOTE If the KB article ID is not known, enter **99999**, and include this information in the posting sub-task to Support Knowledgebase.

How to update old-style links

- You need to enter the attributes as described above; and for Support and PCS links, you need to unwrap or delete the Hypertext element. For Support links, the URL may have the article ID in it, so don't delete too quickly.

What if I am in an agent article and need to use references to both Support and PCS?

What if, indeed. When writing a troubleshooting flow, you may want the agent to direct the customer to an article in Support to do a procedure or check reference information (e.g., supported accessories or email settings for different carriers). You may want the agent to direct a customer to a Support article the agent is using online so the customer can step through a procedure with the agent. In the agent-only article, you need to provide both a Support article reference to be given by the agent to the customer, and a linked reference for the agent to follow. Provide the agent reference first. Use the format below.

- Note in the second example, a phrase is provided to provide context for the reference (the article has multiple sections).
- Not all references need to be provided in both formats. These guidelines are for cases where it makes sense (see 484, 485 for examples).
- If you are not providing references for both the customer and the agent, you do not need to use the "In Solutions..." intro phrase.

Example 1

In Solutions, see **SSL certificates** [356].

Refer the customer to **kb.palm.com**, article ID **47857**, "SSL certificates," for the pre-installed list of certificates.

Example 2

In Solutions, see **Using Screen & Lock** [331] and **Restart/erase the device** [272].

For information on using a PIN or password to lock the phone, refer the customer to **kb.palm.com**, article ID **45249**, "Screen preferences and settings."

For information on remotely erasing the phone, refer the customer to article ID **44616**, "How to remotely erase your phone."

KB writing style

KB articles are not just UG topics packaged as separate articles, although many begin that way and some stay that way. Even articles pulled from a UG must be assessed and rewritten for use as a KB topic. UG topics typically have more cross references that KBs use, and the ones we use have to be rewritten to the relevant KB. The same topic as a KB may have additional information that is not suitable for a UG. Agent troubleshooting flows are a different challenge altogether.

KB style variance from UA

Emphasized words: KB style follows UA style with a few situation-dependent additions. In the interest of usability and scannability. KB articles sometimes use bold or ALL CAPS to draw attention to a word or phrase that is important, such as a file name the customer needs to read online and then locate in a folder on his/her desktop PC. You want the

customer to be able to look down, forget what the word was, and look back and find it easily. If you've ever done this, you know what I'm talking about. If you haven't, you will. Consider it an exception, but if your customer must remember **xlstuiorbz.pda**, you have a tool. Examples of this are to be found in various articles. Usage is contextual, used sparingly, and decided on a case-by-case basis.

ALL CAP usage occurs most commonly in Policy articles, which used this style on a regular basis before these articles became part of the overall KB effort. It was retained there to ease the transition to a newly formatted and edited document set for agents, before it was known agents had never read the old documents. The style was adopted sparingly for KB articles in general because it was useful, and lots of bold becomes hard on the eyes and means less.

Scannability: This applies mostly to agent articles, but not entirely. Troubleshooting articles are written to used by Support agents in real time while they are on a call with a customer. To make an article usable by an agent, information needs to chunked into visual units. Avoid long paragraphs, or too many paragraphs in a row. When you see troubleshooting flows that don't do this, don't assume it's OK. The first batch of 400 or so articles was written in wartime conditions with the emphasis on quantity, with writing and style guidelines developed as we went.

KB articles do not need to be scannable in the same way, but you should keep in mind that these articles are read online, and may be read on a small screen. Try not to write articles that will require lots of scrolling because of lots of screenshots or long procedures. To this end, KB procedures may sometimes combine related actions in one step that the UG would split into two or more. It's more important to be concise in KBs than in the UG, where you have the option to link endlessly to other information and are not constrained as much by length. (Some KBs may be long, but they are typically chunked into short related topics. Some may be just too long, because it can be easier to write long than short.)

Title style

KB how-to articles use a casual style, and generally use the gerund form: "Using titles." This title style doesn't apply in all cases, however, and you'll see other forms. Some older articles still use the style copied from the UG: "Use titles." Fix these when you see them.

Troubleshooting articles are generally symptom oriented and vary more. The KB part of a troubleshooting article is more personable: "I can't find my webOS title." The agent side of this article would be titled something like: "Can't find webOS title."

Subsections use the form "Send email" vs the gerund form "Sending email." KBs originally used a gerund form in subsections, but editorial changed this along the way. Some older articles may still have this style.

Voice and audience

KB articles use the casual UA style. See the UA style guide. Remember that KB articles are written for customers, and agent articles are written for Support agents in call centers. Agent articles don't need to be friendly; they need to be to the point. You can assume the agent has some familiarity with the product, even new products.

Writing troubleshooting articles

Agent side: Troubleshooting flows all use to begin with checking for a software update (INCLUDE UPDATE) unless they cannot (e.g., the issue includes no connectivity). This proved unworkable and was finally stopped. New and revised articles do not begin with the software update step unless that step specifically applies to the solution. Old procedures may still have the step. Write it out of the procedure when an article gets revised.

After they were written, many troubleshooting articles were rewritten for bugs multiple times, adding more and more logic steps and paths, until they were difficult to follow and use. As we come across these articles again, we're assessing them to see how/if they can be broken down into multiple simpler articles addressing specific symptoms.

The logic is critical. Every step must lead to the next. There can be no logic dead ends. Every question and answer must connect. Every path that leads to the end of the call must account for everything. This is not the case in older articles. Not all were edited, and the use of the generic "Did this solve the problem?" allowed a lot of unwritten but necessary steps to be lumped under "this."

Beyond this general guidance, if you need to write one, ask one of the experienced writers to recommend an existing article that might be a good starting point.

It's easy even for experienced writers to write logic that doesn't quite work. Cathi Skyles is great at finding logic holes.

Customer side: There appear to be two distinct schools of thought here. I (Scott) always write the agent side first, then create a customer version. Cathi believes it is easier to write the customer side first. So basically, whatever works for you.

The customer version does not include branching paths and jumping around. It's linear interpretation of the agent troubleshooting article. You typically want to reduce the path to a procedure of the form "Try these steps in this order. Stop when you succeed." If the customer gets to the end of the procedure and still has a problem, the last step is understood to be "contact your wireless service provider," which will head into the more complex agent troubleshooting flow.

Caveats: There are some agent articles that are not troubleshooting flows. They are the rare exception. Some KB troubleshooting articles take the form of FAQs or symptoms and explanations (the form used in UGs). SME input will be all over the map; we have to bend into one of our recognizable forms. SME input is typically a form of simplified troubleshooting flow, which we must fill out and then create a customer side for. Or it may be a simple standard type procedure from which we have to create a troubleshooting flow.

Writing KB how-to articles

KB how-to articles created from UG content are not plug-and-play. The UG has different structure elements, different hierarchy, and a different writing style. UG articles need to be massaged to some extent to become KB articles. Note the following:

- KB image files are smaller than UG images and located in a different source folder. If UG images are to be used, they need to be resized and put into the KB Final_Graphics folder in SVN and then imported into the KB. KB images have to be resized. If this is not done, images will display OK in FrameMaker but not in the HTML.
- UGs make extensive use of conditional text. KBs cannot use any conditional text. Conditional text you don't want to keep must be deleted. All text that you use must be made unconditional. Turn on all condition text to see what the variations say. You may need to rewrite them all into one all-purpose paragraph.
- A single subject in a UG (e.g., "sending email" may be covered in a number of different articles presented at the same level (h1). A KB topic on sending email would combine these heads as subtopics in one article called "Sending email."
- UG articles commonly use more references than a KB. A UG article may refer to topics anywhere in the 400-page UG. A UG article may refer the user many times within a web of related articles (email again). KBs use fewer references and only those that relate closely to the task at hand. You don't want the user to have to jump around more than absolutely necessary.
- KB articles are more standalone. A UG procedure may refer the user to another procedure to get the information required to complete just one step or part of one step (OMG!). In a KB, make the procedure steps complete.
- Along the same lines, if an article contains multiple subtopics, and when displayed on a computer screen, they all can't be seen at once (or with a little scrolling), create an article TOC using a bullet list of cross-refs to the section titles. Use this style for FAQs.
- UGs written for Sprint had a different style from other palm UGs. This has changed now, but you may still come across these differences. More items are bolded and capitalized. leadins are bold. Titles use title cap instead of sentence cap. All these differences have to be returned to Palm style in the KBs. This is important because the Pixi user guide is a Sprint manual. Examples of Sprint style:
    - Cross references may be to "Heading & Page" instead of "Heading"
    - All headings are initial capped—all words of 4 or more characters
    - List items have periods, Palm lists don't
    - In an item in a numbered list, sentences after the first sentence are in parenthesis.
    - Sprint uses Web instead of web.
    - Sprint uses **Application** menu instead of **application menu**. In general most menus and buttons and UI elements are bold—many than in Palm style or just about anything else you'll see.
    - In bulleted lists that present menu options (usually in procedures), the leadins are bold. Not in Palm style.

Writing/editing Policy docs

Style notes
Many Policy KBs still use the Title cap style instead of the Sentence cap title style. Change this when you see it.

Policy docs historically had no defined style (or all defined styles). They have been made somewhat consistent within themselves and to a degree with other HP docs, but not entirely so. There is no separate Policy style guide or guide for documenting Siebel procedures.

Those who provide input for Policy docs tend to the style of capitalizing just about all nouns. You can safely lowercase just about everything. Very few should actually be uppercase (Service Request (SR), Service Repair Order (SRO), Telephone Support Order (TSO), Accessory Repair Order (ARO) are among the few.

Most Siebel procedures were documented without access to screenshots or the application. Most input for Siebel procedures capitalizes and bolds things in a way that may or may not match the UI. There's a need here to bring this together more. Get access to a Siebel account and some actual Siebel documentation to see how they handle things.

Because many Policy KBs present policy information, the input may adopt a pseudo-legal writing style that sounds more official. It's not necessary and can be rewritten to something simple and able to be understood by agents.

Input for policy docs doesn't recognize the difference between procedures and process descriptions and narrative. We have to create those distinctions. Just because input is numbered doesn't mean it's a procedure, etc.

Related topics
Avoid using this section in Policy docs. This is an optional section for use in Support articles. When using, consider implications of links. Are they really important? Important links were likely used within the body of an article. How likely is it that the related articles links will be used. The reason you have to think of these things is that the links need to be accurate for all products and carriers to which the article applies, or you need to provide guidance for updating them. KB articles are generally used by users to solve specific problems or find specific information; not to browse to see what's there. If you edit an article that has this section, and the links are still dummy links, assess as to whether the whole section could be deleted.

Relationship between writer and SME/reviewer

The SMEs and others who request articles to be created/updated and who provide input and review the articles have definite opinions about what needs to be included. You can figure that they'll know more than you do about how things work—and you need to respect that—but remember that you are the professional technical writer. They know what they want. They may not know what is correct for the user. SMEs may provide text that is too technical, legally suspect, uses internal-to-Palm/HP terminology, or just plain doesn't make sense. They may have the right information but be asking for the wrong

article to be updated. It's your job to recognize and fix all of that. Be discerning and all knowing.

Siebel codes:

- Level 1: Required for all AT and KB articles but not Policy KBs
- Level 2: Check PCS for these for existing articles
- Level 3: And then your SME

HP App Catalog download issue

INCLUDE REMINDER:

IMPORTANT HP supports finding, purchasing, downloading, installing, updating, and deleting App Catalog applications. The developer cannot assist the customer with any of these functions. **Policy** [542]

1 Ask the customer if the app is being downloaded through App Catalog.

YES
Continue.

NO
Explain to the customer that HP supports only apps that are downloaded through App Catalog. HP cannot assist with apps that are downloaded from other sources.

INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

2 Confirm that the problem has to do with downloading an app from App Catalog and not another issue. Use these guidelines:

- **Problems finding:** The customer cannot find an app in App Catalog or the customer is having problems locating the link for a beta or web-distributed app.
- **Purchase or payment issues:** The customer received an error when trying to purchase an app, or there is an issue with the payment or charges.
- **Problems downloading:** The customer cannot download an app or find a downloaded app, received a "Can't Install" message indicating insufficient space, or received a message indicating the app is not compatible with the customer's device.
- **Problems installing:** The customer saw one of these two error messages: "Couldn't install" or "Install failed."

Which best describes the situation?

PROBLEMS FINDING
Go to **Trouble finding an app in App Catalog** [545].

PURCHASE OR PAYMENT ISSUES
Depending on the situation, go to one of the following articles:

- **App Catalog payment issue** [544]
- **Using HP App Catalog promo codes** [719]
- **Setting up and using carrier billing in HP App Catalog (AT&T)** [723]
- **Using App Catalog** [170]

PROBLEMS DOWNLOADING
Continue.

PROBLEMS INSTALLING
Go to **App Catalog installation issue** [172].

3 Verify that the customer has a data connection, either from the wireless service provider indicated by the 1x, 3G, E, Ev, G, H, H+ or U icon, or via Wi-Fi, indicated by the icon (with two or three bars). Note that a tablet device may be capable of both Wi-Fi and carrier wireless connectivity, or only Wi-Fi; depending on the model.

Does the customer have a data connection?

YES
Continue.

NO
Offer these tips for improving the data connection:

To get a better Wi-Fi signal:

- Move closer to the Wi-Fi source.
- Move the Wi-Fi access point off the floor and away from walls and metal objects.
- Move the device away from other wireless devices such as cordless phones, microwave ovens, baby monitors, etc.
- If you have no Wi-Fi connection, see **Can't establish a data connection using a Wi-Fi network** [133].

To get a better wireless service provider data connection:

- If you're standing, move about 10 feet in any direction.
- In a building, move near a window. (Open any metal blinds.)
- In a building, move out of doors or to a more open area.
- Outdoors, move away from large buildings, trees, or electrical wires.
- In a vehicle, move your device so that it's level with a window.
- If you have no data connection, see **Can't establish a data connection using a wireless service provider's network** [115].

Can the customer obtain a good data connection?

YES
Continue.

NO
Explain that a data connection is required to fully troubleshoot this issue online.

INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

4 Ask if the customer followed a link to an app and received one of these errors:

- **This application is not available for your model.**
- **This application is not available for your carrier.**
- **This application is not available in your country.**
- **This application cannot run on your current operating system. Use the Updates app to install the system update and try again.**

- **This application cannot be installed on your device.** If multiple violations apply, this general message may appear.
- **This is not a valid webOS application.**

YES
Explain that the apps available for a device vary according to webOS version, model, and country. Because customers can send links to apps to each other, either by using the Share button in App Catalog, or directly from an Internet site, it's possible for a customer to receive a link to an app that isn't supported on the customer's device. If the customer taps the link, one of these error messages displays.

INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure.

NO
Continue.

5 Ask if the customer received one of the following messages:

- **Can't Install** message indicating some apps or files need to be deleted before the install.
- **Download Error** message indicating the download file is missing or damaged.
- **Can't Download** message indicating the United States government prohibits the download.
- **Other** error.

YES

Which error message did the customer see?

CAN'T INSTALL
If the error message was about deleting apps or files, advise the customer to make space for the new application by removing:

- Applications that are no longer in use **How to** [170]
- Files from the USB drive **How to** [273]

Did this solve the problem?

YES
INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

NO
Continue.

DOWNLOAD ERROR
If the error message was about a missing or damaged download file, advise the customer to follow the prompts to fix the issue.

Did this solve the problem?

YES
INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

NO
Continue.

CAN'T DOWNLOAD (PROHIBITED BY U.S.GOVERNMENT)
If the error message indicates that the United States government prohibits the download, escalate the case immediately. **Policy** [264]

INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

OTHER
Ask the customer to follow the prompts for the error and attempt to download the app again.

Did this solve the problem?

YES
INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

NO
Continue.

NO
Continue.

6 Ask the customer to open **Software Manager** and look for the app.

Is the app there?

YES
Continue.

NO
Go to step 8.

7 Ask if the customer sees any of the following icons: pause , error , new version (smartphone) or (tablet), or resume .

YES

Which one?

PAUSE
Explain to the customer that the app is still downloading and that there is no indication of a problem. Ask the customer to allow the download to finish.

INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

ERROR

Ask the customer to tap the icon and follow the prompts to fix the issue.

Did this solve the problem?

YES

INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

NO

Continue.

NEW VERSION

Explain to the customer that the application is already on the device and that a new version is available. Help the customer find the app in the Launcher and explain that the customer can return to this screen to download the new version when convenient.

INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

RESUME

Explain to the customer that the download is currently paused. Ask the customer to tap the icon to continue downloading the app.

INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

NO

Explain to the customer that the app has finished downloading. Assist the customer with finding the app in the Launcher. **How to** [311]

INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

8 Ask the customer to restart the device. **How to** [272]
After the device restarts, ask the customer to find the app in App Catalog or tap the link for the app (if this is a beta app or web-distributed app) and attempt the download again.

Did this solve the problem?

YES

INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

NO

Continue.

9 Ask the customer if he or she previously installed an app from a source other than App Catalog (these apps are sometimes referred to as “homebrew” apps).

YES

- Explain to the customer that HP supports only apps that are installed through App Catalog. HP cannot assist with apps that are installed from other sources.

- Because these applications can impact the system in many ways, offer the customer help with restoring the device to its factory settings using webOS Doctor. **How to** [222]

INCLUDE SOLVED: Update the case notes, and close the case following the standard procedure. **Policy** [294]

NO

Explain to the customer that HP supports finding, purchasing, downloading, installing, updating, and deleting App Catalog applications. The developer cannot assist the customer with any of these functions. **Policy** [542]

The customer can report the problem to HP's App Catalog team:

a Open **Software Manager** .

b **Apps** is selected by default, showing downloaded apps that appear in the Launcher. Tap **Other** to manage background apps and services that do not appear In the Launcher.

c Tap the app or service name that is causing the problem.

d On the app details screen, tap **Report A Problem**. Under Problem Type, tap **Bug** and tap a problem type to narrow the type of problem you are reporting. Enter the problem description, and tap or **Send**.

INCLUDE NOT SOLVED: Update the case notes, and escalate the case following the standard procedure. **Policy** [264]

I'm having trouble downloading an application through HP App Catalog

NOTE You can install applications to your webOS device from sources other than HP App Catalog. These apps are sometimes referred to as "homebrew" apps. Homebrew apps have not been tested and certified by HP. HP cannot assist with the finding, purchasing, downloading, installing, or removing of applications that are not installed through App Catalog. For more info on homebrew apps, see Using third-party applications from sources other than HP App Catalog [85477,].

Try these steps if you can't download an application using App Catalog. Stop when the problem is solved.

1. If you have a version of the app installed without using App Catalog, delete it before installing the App Catalog version. All applications have a unique App ID, even those that did not originate in App Catalog. Your webOS device can serve only one App ID instance at any given time.
2. If you received an error message that there is not enough room to install the app, make space for the new application by deleting applications you don't use or moving to a computer files you don't use. See "Deleting applications" in Using HP App Catalog [22785,] or Using USB drive mode to transfer files to your device [50331,].

IMPORTANT Information stored in an application is stored only within the application, so you cannot access this info after you delete the app. Before you delete an app, write down any info you want to save. Alternately, you can save a screen capture of relevant info by pressing and holding **Option** + **Sym** + **P** (smartphone) or the **power** + **center** buttons (tablet). Screens you capture are saved in the screencaptures folder.

> TIP If you delete an app purchased from App Catalog, you don't have to purchase the application again. Purchased applications are associated with your HP webOS Account.

3. Look for the app in Software Manager.
   a. Open **Software Manager** .
   b. **Apps** is selected by default, showing downloaded apps that appear in the Launcher. Tap **Other** to manage background apps and services that do not appear In the Launcher.

   You may see one of the following icons:

Icon	Description
	The download or installation of the app failed. Tap the icon and follow the prompts to fix the issue.
	The app is currently downloading.

Icon	Description
	The app download is currently paused. Tap the icon to resume the download.
1.0.30	A version number indicates the application is already on your device. If there is a down arrow next to it, an update is available for the app.

4 On smartphones or tablets with a wireless service provider plan, check for a roaming icon. If you are roaming, check to see if data roaming is enabled (additional charges may apply; check with your wireless service provider), or go to step 5 to try the download using a Wi-Fi connection (if Wi-Fi is available on your device).

To enable data roaming:

a Open **Phone** .

b Open the **application menu** and tap **Preferences**.

c Under Network, set the following:

- **Voice Network:** Set to Automatic to allow roaming on other networks.
- **Data roaming:** Set to Enabled to allow web browsing, email, multimedia messaging, and instant messaging when you are roaming. This option appears only if you have set Voice Network to Automatic.
- **Data Usage:** Set to ON to enable data usage.

5 Establish a Wi-Fi connection (if Wi-Fi is available on your device), and then try the download again. See Using Wi-Fi [21771,].

6 Restart your device, and then try the download again:

a Open **Device Info** .

b Tap **Reset Options**.

c Tap **Restart**.

d After the device restarts, find the app in App Catalog, or tap the link for the app (if this is a beta app or web-distributed app) and attempt the download again.

Siebel codes:

- Level 1: Using third-party software
- Level 2: Finding and downloading
- Level 3: Third party download or purchase issue