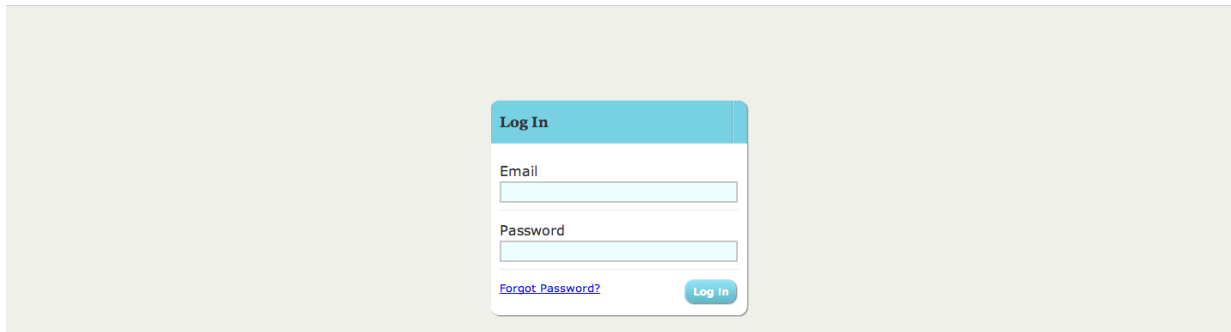


# Flow 1.0 Heuristic Evaluation Report

---

rhythm



Darold Davis  
User Interface / User Experience Engineer

---

July 23, 2013

Revised 18 September 2013



## EXECUTIVE SUMMARY

---

This report provides a detailed analysis of the heuristic evaluation process used to evaluate Flow 1.0, Rhythm's publishing application. The evaluation itself was performed using the heuristic evaluation usability method, based on heuristics provided by Jakob Nielsen. This method consists of evaluators comparing a pre-defined set of usability principles to an application or website while attempting to complete a system task.

For this project, eight heuristics were used, focusing on the core functionalities of Flow: setting up a creative, generating reports and media planning. The goal of this evaluation was to identify major usability flaws within the Flow interface through the application of these nine heuristics:

1. Aesthetic and minimalist design
2. Match between system and real world
3. Recognition rather than recall
4. Consistency and standards
5. Visibility of system status
6. Flexibility and efficiency of use
7. User control and freedom

The usability problems found from this evaluation are clustered into eleven areas and are ranked according to their severity and ease with which the problem can be solved. The seven most severe and easiest to fix problems are:

1. **Inconsistencies between icons and labels**
2. **Some actions have "no way outs", or way to undo**
3. **Buttons and hyper link interactivity are not salient**
4. **Accelerators in contextual spaces are lacking, causing inefficiencies**
5. Some inconsistencies with browser standards
6. **User is not kept informed of the systems status**

The last four usability problems that are not discussed in detail in this report are: "Help content uses different terminology than application," "Default text in interface is difficult to read," "Similar buttons in different areas do different things," and "System does not always provide user with enough information about the task being performed." These problems are not discussed in detail because evaluators have classified them as superficial usability problems that should be fixed only if extra time is available.

## INTRODUCTION

---

This report describes the heuristic evaluation process and findings for Rhythm's Flow publisher application. It begins with summary information describing the product and its target population, and then continues with an overview of the heuristic evaluation technique in general, the specific goals of this project, and the heuristics used in my cognitive walk-through of the application. Following is a summary of the major problems identified by the heuristic evaluation and a more detailed exploration of the findings related to these specific problems prioritized according to their severity and impact on the user experience. The report concludes with a list of resources related to heuristic evaluation and a detailed list of the actual heuristics used in this evaluation.

## PRODUCT INFORMATION

---

### **Product Description**

Flow is Rhythm's unified, comprehensive technology platform for generating higher tablet and mobile revenue. Flow powers direct sales and serving of brand advertising, ad-supported syndication & distribution, and brand-focused audience targeting. Flow was designed to help publishers build real, significant tablet and mobile businesses and has proven its monetization prowess over the last three years for publishers such as E!, CBS/TV.com, CNBC, TMZ, Demand Media, and IAC.

Flow is also the same platform powering Rhythm's direct advertising sales, and of course Flow natively supports the option of using Rhythm as a complementary premium sales channel. More can be learned from the Rhythm website - [www.Rhythm.com](http://www.Rhythm.com).

### **Target Population**

Flow is targeted to external facing publishers and their Ad Ops teams using computers running Chrome, Safari and Firefox. In addition, Rhythm's internal Ad Ops is part of the target population. There are no specific demographic targets identified.

Home Campaign Trafficking Media Planner Creatives Ad Units & Packages Publishers & Apps Advertisers & Agencies Reports Admin

Filtering by active

Home > Insertion Orders > New Insertion Order Save IO

**New Insertion Order [generic] <sup>2</sup>**

Campaign Name	<input type="text"/>	Start Date	<input type="text"/>	Advertiser	<input type="text"/>	Seller	<input type="text"/>
IO Number	RNM-204954D3D3271	End Date	<input type="text"/>	Agency	<input type="text"/>	Account Manager	<input type="text"/>
Comment	Creating new IO	Ad Category	<input type="text"/>	Sold by publisher	No	Campaign Manager	<input type="text"/>

Last Modified By: Darold Davis

# Line Items: Total Budget: Total Reserved: Total Served: Progress:

**Line Items (0)** (ordered by Placement Name)

**Exclusion Rules**

Save & Close IO | Add a Line Item [ [Standard](#) | [Roadblock](#) | [House](#) | [Off Platform](#) ] or [Add an Exclusion Rule](#) | [Cancel](#)

Copyright © 2013 Rhythm NewMedia. All rights reserved. Version 7.8.60522 [Submit Feedback](#) This page was last refreshed on 07/11/2013 08:02 PM EDT (Ad Server time)

Figure 1: Flow Interface

## HEURISTIC EVALUATION TECHNIQUE

---

### Methodology

The heuristic evaluation usability method was used to produce the findings summarized in this report. According to renowned usability expert Jakob Nielsen, a “heuristic evaluation involves having a small set of evaluators examine the interface and judge its compliance with recognized usability principles (the ‘heuristics’)” (*How to conduct a heuristic evaluation*). Commonly referred to as a “discount” usability technique (Nielsen, 1993, p. 160), this method allows evaluators to discover possible usability problems in a product or application in a single afternoon. Later, more expensive and extensive user testing can investigate the usability problems identified through heuristic evaluation. When conducting a heuristic evaluation, evaluators compare a pre-defined set of specific usability principles with a product or web site interface while attempting to accomplish actual system tasks. Evaluators may either work individually; combining findings later, or they may perform the evaluation at the same time with each individual focusing on several different heuristics.

### Specific Project Goals

For this project, I developed and used a set of eight heuristics to discover usability problems in the Flow digital music application interface. This evaluation focused on the core functionality of the Flow interface: Campaign, Media Planner, and Reports. The interactions with Flow’s other features were not investigated. I prepared results for two heuristics individually, and then evaluated the remaining heuristics in a product cognitive walk-through, summarized in a separate report. Larger problem areas and severities were reached through group consensus.

The goal of this project was to identify major usability flaws in the Flow digital publishing application using established heuristic evaluation techniques. In addition, the heuristic evaluation revealed controversial situations where the application may violate traditional heuristics while remaining usable for the actual user. Once these usability problems were identified, they were prioritized. Selected areas will be investigated further through user testing.

### Heuristics Used

Many of the most commonly available lists of heuristics are specifically oriented toward the evaluation of static web sites. As a web application, Flow has a different interface and navigation style than a traditional web site. To best accommodate these needs, I combined a set of ten heuristics suggested by Jakob Nielsen (1993) and a set of nine design principles suggested by Saul Greenberg, a professor of Human-Computer Interaction at the University of Calgary (*Design Principles and Usability Heuristics*). To aid in the evaluation process, the broad heuristics were supplemented with more descriptive notes and examples borrowed from Olson (lecture notes, February 10, 2005). The nine heuristics used to evaluate the Flow application are summarized below and listed in full detail in Appendix A.

Number	Broad Heuristic
1	Aesthetic and minimalist design
2	Match between system and real world
3	Recognition rather than recall
4	Consistency and standards
5	Visibility of system status
6	Flexibility and efficiency of use
7	User control and freedom

### Prioritization of Problems

In order to usefully group the findings resulting from this heuristic evaluation process, I clustered specific instances of heuristic violations into eleven problem areas. To further understand the impact of each of these problems, I estimated both its severity in terms of usability principles and the ease with which the problem might be solved. Problem severity ratings were impacted by the frequency with which the problem occurred, the ease with which the user could overcome the problem, and the persistence of the problem—whether it could be solved once or would bother the user every time a task was attempted. This resulted in a dual rating for each problem found, which was used to prioritize the problem areas for presentation in this report. The tables below define the severity and ease of fix rating systems applied. Severity ranks are based on those defined by Jakob Nielsen (*Severity ratings for usability problems*).

Severity Rankings	
Rating	Definition
0	Violates a heuristic but doesn't seem to be a usability problem.
1	Superficial usability problem: may be easily overcome by user or occurs extremely infrequently. Does not need to be fixed for next release unless extra time is
2	Minor usability problem: may occur more frequently or be more difficult to overcome. Fixing this should be given low priority for next release.
3	Major usability problem: occurs frequently and persistently or users may be unable or unaware of how to fix the problem. Important to fix, so should be given high priority.
4	Usability catastrophe: Seriously impairs use of product and cannot be overcome by users. Imperative to fix this before product can be released.

<b>Ease of Fixing Rankings</b>	
<b>Rating</b>	<b>Definition</b>
0	Problem would be extremely easy to fix. Could be completed by one team member before next release.
1	Problem would be easy to fix. Involves specific interface elements and solution is clear.
2	Problem would require some effort to fix. Involves multiple aspects of the interface or would require team of developers to implement changes before next release or solution is not clear.
3	Usability problem would be difficult to fix. Requires concentrated development effort to finish before next release, involves multiple aspects of interface. Solution may not be immediately obvious or may be disputed.

---



## SUMMARY OF FINDINGS

---

After completing the heuristic evaluation of the Flow digital publishing application, I identified eleven problem areas that violate traditional usability principles. These problems have been prioritized below, with the most severe and easiest to fix problems listed first. It illustrates that Flow has the largest number of usability problems in the area of consistency – this heuristic was violated in six of the eleven problem areas. Particularly for Web users unfamiliar with the traditional Rhythm interface elements, there may also be significant memory load problems (Heuristic #4 – Minimize user’s memory load). While these areas may not be considered problematic by Rhythm’s own AdOps users, external AdOps users in publishing companies are part of the Flow target population. To better support this part of their target population, Flow developers may need to add additional customization to the new version of Flow. Of the eleven identified problem areas, the seven most severe will be discussed in more detail in this report.

#	Problem	Severity Ranking	Ease of Fixing Ranking	Heuristic Number	Broad Heuristic
1	<b>Inconsistencies between icons and labels</b>	3	1	#5	Be Consistent
2	<b>Buttons and hyper-links interactivity are not salient</b>	3	2	#1, #4, #5	Aesthetic and minimalist design; Minimize user’s memory load; Be consistent
3	<b>Accelerators in contextual spaces are lacking, causing inefficiencies</b>	2	1	#4	Minimize user’s memory load
4	<b>Some actions have “no way outs”, or way to undo.</b>	2	3	#7	Provide clearly marked exits
7	Modal interface causes inconsistency in available features.	2	3	#5	Be Consistent
8	Help content uses different terminology than application.	1	0	#5, #9	Be Consistent; Provide Help
10	<b>User is not kept informed of the systems status</b>	3	2	#1, #5	Visual Feedback, Be Consistent
11	System does not always provide user with enough information about the task being performed.	1	3	#6, #8	Provide feedback; Deal with errors in a positive manner

## SPECIFIC PROBLEM AREAS

---

### 1. Inconsistencies between icons and labels

#	Problem	Severity Ranking	Ease of Fixing Ranking	Heuristic Number	Broad Heuristic
1	Inconsistencies between icons and labels	3	1	#5	Be Consistent

#### Problem:

While analyzing the interface, I noticed inconsistencies between language used in the labels, buttons, and tooltips available within the interface. These problems violate heuristic #5, which states that vocabulary, labeling, and functionality should be consistent within specific tasks and across the interface as a whole. This problem area is ranked as a major usability problem because it occurs in various places throughout the interface and also because it cannot be solved by any specific action on the part of the user. Instead, the user must learn to associate different terminology with the same task, increasing the amount of time required to learn the task. In addition, users may simply become confused by this variant of terminology and icons. If the menu bar and the contextual right-click menus are not consistent in the options offered, users may be fooled into believing that a specific task cannot be performed.

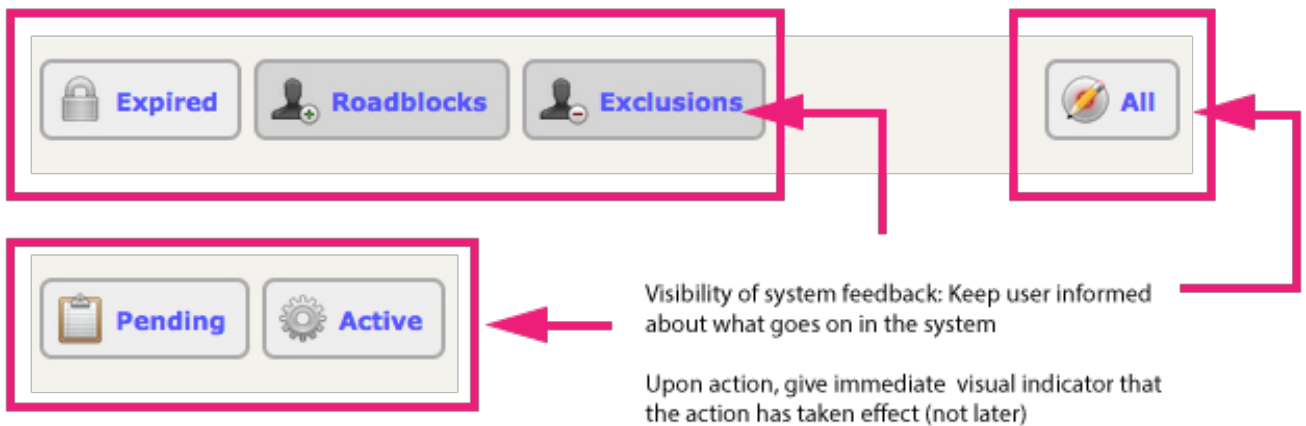
#### Evidence:

This problem occurs throughout the Flow interface. It would require an even more complete evaluation by designers and usability professionals to discover all inconsistencies. However, specific examples of varying language were noted by myself:

Icon	Label
User profile icon	Roadblocks
Clipboard icon	Pending
Gear icon	Active

The table above captures only a few of the icon / label association that presents a mismatch in meaning. In the figure below, the clipboard icon is associated with the label "Pending", which is not consistent.

In addition, other icons such as the gear icon for "Active" also present the same semantic problem.



**Recommendation:**

The obvious and simple solution to this problem is to perform a vocabulary analysis for this application. Specifically, terminology used for buttons and tooltips in the application interface should be compared with terminology used for the same functionality found in similar interfaces and competing products. Once a common language has been identified across various interfaces, I can establish the desired language for each function and then implement that language throughout all aspects of the interface.

2. Buttons and hyper link interactivity are not salient

#	Problem	Severity Ranking	Ease of Fixing Ranking	Heuristic Number	Broad Heuristic
2	Some language does not correspond with user terminology.	3	1	#3	Use Simple and Natural Dialog

**Problem:**

This problem is ranked as a major usability problem because it impairs the user's awareness and use of existing Flow functionality. When exploration is the means to discovery of the interface, the user can become lost and hindered because some links or buttons functionality are not immediately apparent.

**Evidence:**

The use of language that does not correspond with user terminology is not a widespread problem in Flow. However, there are several specific, important instances where unusual language may obscure the completion of routine tasks. A user looking click on a link from a tabular list of items, may assume that nothing is interactive, as the items are not consistent with web interface

heuristics. It was only by accident that I discovered that clicking on the names of campaigns that opened another view.

Another situation in Flow where this is problematic is the very landing page. Though the user could “poke around in the dark” to discover what elements are interactive, the design of those elements are not prominent in contrast to its environment.



The screenshot shows a navigation bar with the following items: Home, Campaign Trafficking, Media Planner, Creatives, Ad Units & Packages, and Publishers & Apps. Below the navigation bar is a breadcrumb trail: Home » Apps. A table with two columns, NAME and APP ID, is displayed. The table contains the following data:

NAME	APP ID
<a href="#">Play Festival Films iPad</a>	<a href="#">playfestfilms_ipad</a>
<a href="#">Play Festival Films iP...</a>	<a href="#">playfestfilms_iphone</a>
<a href="#">45 Sound iPhone</a>	<a href="#">45sound_iphone</a>
<a href="#">48 Hours iPad</a>	<a href="#">48hours_ipad</a>
<a href="#">ABC Family MW HTML5</a>	<a href="#">abcf_mwhtml5</a>
<a href="#">ABC Family Mobile Web</a>	<a href="#">abcfamily_web</a>

The screenshot shows a dashboard with a navigation bar containing: Creatives, Ad Units & Packages, Publishers & Apps, Advertisers & Agencies, Reports, and Admin. The main content area displays two statistics: "active campaigns: 187" and "draft campaigns: 81". A red box highlights the numbers 187 and 81. A red arrow points from a text box to the right of the numbers, which contains the text: "No visual indication that this is an interactive element."

**Recommendation:**

Most elements that the user interacts with are presented as some type of anchor links, so using CSS to style buttons for these would be a small matter. Color contrast is another solution that could make the actionable elements more salient. Once again, Flow designers and developers could work together to define more appropriate interaction for these situations. Obtaining input from actual users would also be helpful in identifying the most recognizable interactions. Once decisions were made about specific interactivity, it would be a relatively simple task to change the text used in Flow buttons and menus.

---

3. User is not kept informed of the systems status

#	Problem	Severity Ranking	Ease of Fixing Ranking	Heuristic Number	Broad Heuristic
3	User not informed of system status when states change.	3	2	#1, #5	Aesthetic and minimalist design; Visibility of system status

**Problem:**

A widespread problem encountered throughout the Flow interface is system feedback. Part of this confusion stems from the fact that Flow originated as an internal application for those familiar with the system, but is now being designed for external customers. Choice of color for some visual cues, and lack of any basic visual feedback contribute to these problems. This problem area violates heuristics #1 and #5 (Aesthetic and minimalist design & Visibility of system status), which suggest that a system should visually inform the user of the state / status of activities. This is important both in creating an appealing visual design and in minimizing the user’s confusion / frustration as to system activity. System feedback and behavior should also be consistent with the standards set in other applications available for a specific computing platform. By implementing visual cues and feedback that are obvious, intuitive, and consistent with other programs, Flow designers can help reduce the frustration for new Flow users as well as help make power users aware of system activity in the interface.

**Evidence:**

There are several specific instances where portions of the Flow interface that are clickable buttons “hide” their resulting activity from users. One of the most apparent is that there is no way to tell if any of the links that have been clicked has been executed. This is a system wide issue and impacts usability tremendously. If the system was fast in its response time, this may not be such an issue. However as the system is now, I was confused by the lack of any activity cues—if it wasn’t for the browsers on feedback system, I would have assumed that my actions had no effect. As the images below illustrate, selecting the checkbox does nothing to update the status of Line Items status of paused or stopped.

Visibility of system feedback: Keep user informed about what goes on in the system

Upon action, give immediate visual indicator that the action has taken effect (not later).

The screenshot displays an advertising campaign management interface. At the top, a header reads "Line Items (18) (ordered by Placement Name)". Below this, a table lists campaign details for a specific line item. The table has columns for Placement Name, Start date, End date, Rich Media, Added Value, Ready / Pause, Package, Ad Unit, Billable, Budget, CPM, Contracted Impressions, and Delivered Impressions. The "Ready / Pause" column for the first row contains a checked checkbox. A red box highlights the "Line Items (18)" header, another red box highlights the "Ready / Pause" checkbox, and a third red box highlights the "End date" field. Red arrows point from the "Line Items (18)" header to the "Ready / Pause" checkbox and from the "End date" field to the "Ready / Pause" checkbox. Below the table, there are buttons for "Targeting", "Add Creatives", "Creatives", "Timed Frequency Caps", and "Advanced Options".

Placement Name	Run: 37135	Start date	End date	Rich Media	Added Value	Ready / Pause	Package	Ad Unit	Billable	Budget	CPM	Contracted Impressions	Delivered Impressions
zTEST Android mediamind ad tag test impressions		10/26/2012 12:00 AM EDT	01/31/2013 11:55 PM CET			<input checked="" type="checkbox"/>		Banner	No - Test Line Item	\$10.00	\$1.00	10,000	245

**Recommendation:**

Solving this problem poses a more difficult design challenge in part because it would likely be difficult to find a solution acceptable to all interested parties. The conscious choice to make the Flow interface visualize its activity as much as possible meshes well with a responsive system. However, a better understanding of the backend system would be required to assess what is being called, how long it takes to retrieve, etc. Still, even a basic progress bar (for long page loads) would be sufficient enough.

---

4. Accelerators in contextual spaces are lacking, causing inefficiencies

#	Problem	Severity Ranking	Ease of Fixing Ranking	Heuristic Number	Broad Heuristic
4	Context specific buttons are lacking.	2	1	#6	Flexibility and efficiency of use.

**Problem:**

Making shortcuts and options available in the context of an operation has become a common practice in many graphical applications. These tools generally provide very succinct options of that context's functionality in order to assist users engage more efficiently with the task at hand. Small graphical buttons made available in the context of certain operations can also play the role of reducing the pages or steps a user requires to complete a task. With contextual options, a user does not have to jump out of context, interrupting the flow—they can act while its happening.

## Evidence:

Some events in Flow do indeed have contextual options associated with them. In fact when working with Line Items, options are available that are specific to the scenario at hand. However, the situation in which a user wants to edit (because “edit” is there, although ghosted out) a creative is one in which a contextual option would be efficient. The user is presented with an alert that the campaign must be paused before editing the creative. To do this, the user has to leave the context that they are in and go to a different page to pause the campaign. This is especially problematic because now the flow and focus is broken and the user has to search for the campaign associated with the creative and then pause it.

## Recommendation:

To solve this problem, a pause button could be added next to the edit button in the Flow interface, thereby making it immediately efficient to pause then edit the creative in context. This task would not be difficult for a developer to accomplish.

Flexibility and efficiency of use: Accelerators should be provided.

Provide pause button in this context

rhythm

Publisher Campaign Manager

The screenshot shows the Rhythm Publisher Campaign Manager interface. The top navigation bar includes 'Home', 'Campaign Trafficking', 'Media Planner', 'Creatives', and 'Agencies'. The 'Creatives' tab is active, and a modal dialog box is displayed over the 'edit' button. The dialog box contains the following text: 'The page at https://publisher.rnmd.net says: You must pause the campaign or pause all associated line items before editing the creative!' and an 'OK' button. The background interface shows the details for a creative named '20130723 AMC Breaking Bad S6 JESSE Banner'. The details include: NAME: 20130723 AMC Break, AD UNIT: Banner, DESCRIPTION: (empty), CREATIVE TYPE: Image, TRACKING URLS: 1 http://ad.doubleclick.net/ad/N4427.127453.RHYTHMNEWMEDIAINC/B7778765.3;sz=1x1;ord=[timestar], CLICK THROUGH ACTION: Click Through URL, CLICK THROUGH URL: http://ad.doubleclick.net/clk;274670453;101042222;n, and TRANSPARENT LANDING PAGE: No. A preview of the creative is shown at the bottom, featuring the text 'Breaking Bad' and 'RETURNS AUG 11 9/8C AMC TAP FOR VIDEO'.





---

## 6. Undo commands basically unsupported

#	Problem	Severity Ranking	Ease of Fixing Ranking	Heuristic Number	Broad Heuristic
6	Undo commands basically unsupported.	2	3	#7	Provide clearly marked exits

### Problem:

As a rule, general computer users are accustomed to being able to back out of a change or mistake they make by selecting some type of UNDO. This command can be found in almost every computer application. Unfortunately the UNDO command has limited support for undoing user actions in Flow. Furthermore, where it *is* supported it exhibits quirky behavior. This violates the seventh heuristic, Provide clearly marked exits, which emphasizes that users should be able to undo or back out of unwanted system states easily. The problem is ranked as minor since the changes and mistakes are easy enough to overcome, but could be simplified more.

### Evidence:

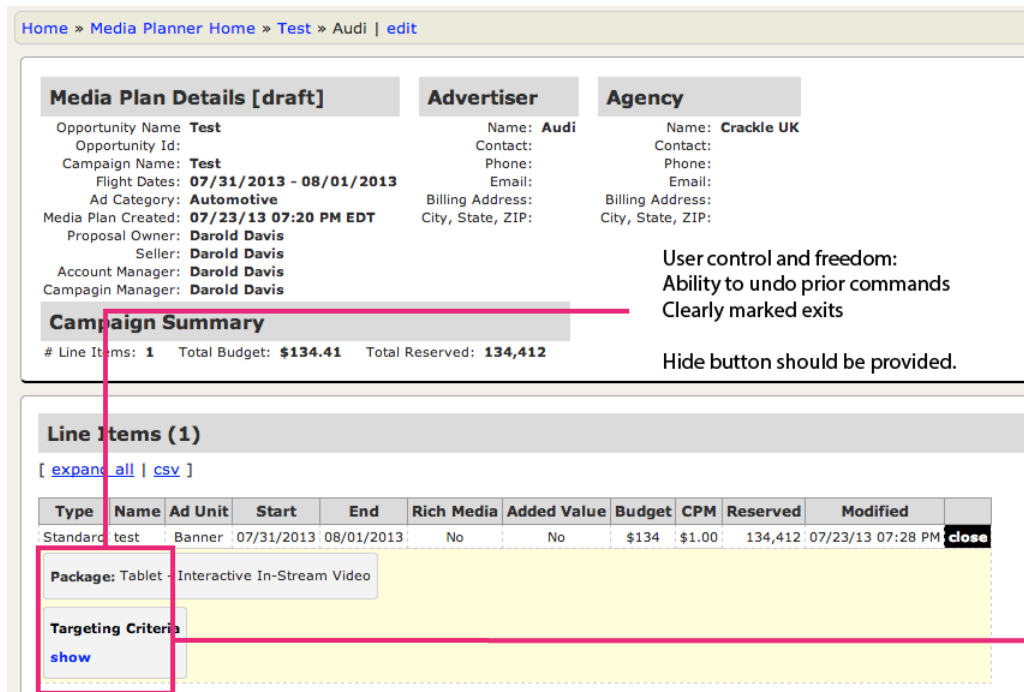
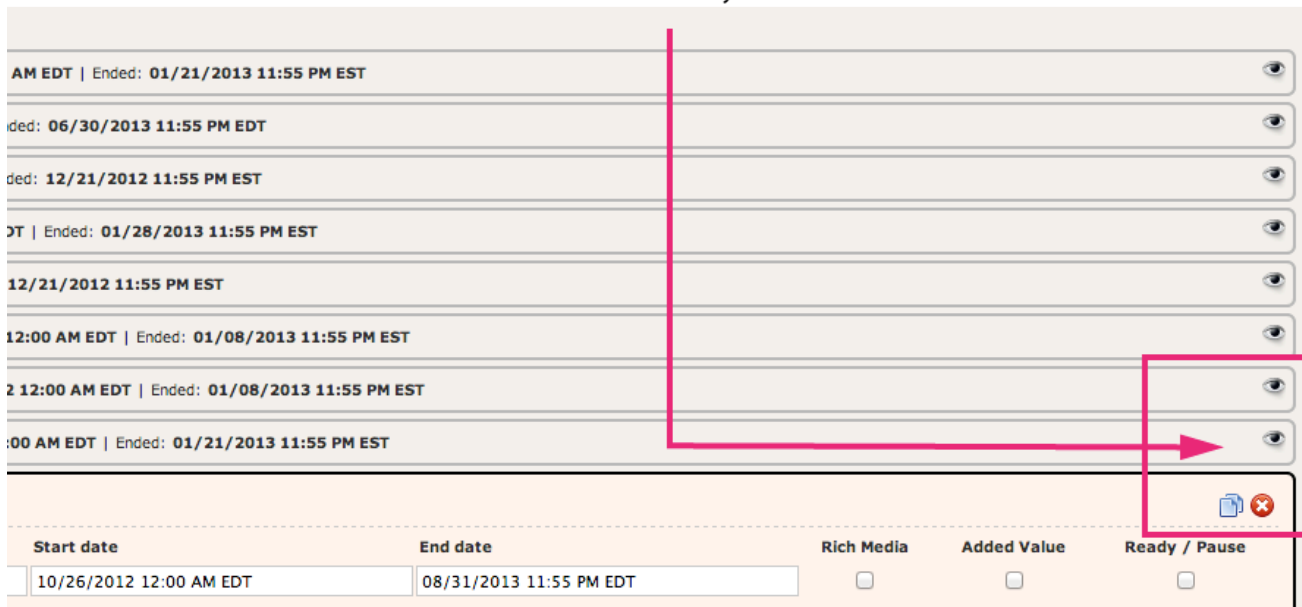
*Unsupported:* When editing an Insertion Order, several other IOs may be listed but closed. A small eye icon indicates that further details may be revealed or shown. However, there is no way to undo this action; to close the IO. When open, there is a red circle icon with an "X" that would seem to indicate some sort of a close action, but this is a fatal mistake. The icon is in fact to delete the IO entirely. The only way to close any open IOs is to refresh the page. In its current state there is no way to undo this action.

### Recommendation:

One obvious and simple solution to this problem is to add the appropriate inverse to open eye icon; a closed eye icon to indicate. Similarly in the Media Plan Detail view, a "hide" option could be added along with the show option that is already there. In addition, undo support for backing out of modifications made to a play list, either through deletion or reordering, should be enabled. The ability for a user to recover from open states (and avoid deleting an IO) without having to refresh the page would greatly enhance the user experience.

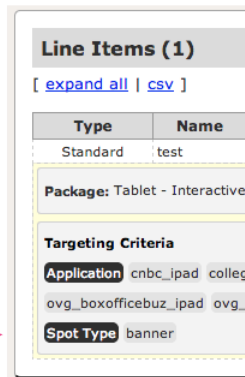
User control and freedom:  
 Ability to undo prior commands  
 Clearly marked exits

Close button should be provided.  
 The red "X" button actually deletes the line item



User control and freedom:  
 Ability to undo prior commands  
 Clearly marked exits

Hide button should be provided.



## SUMMARY

---

While Rhythm's Flow digital music application is generally considered easy to use, a detailed heuristic evaluation based on nine general usability principles revealed a number of specific usability problems. These specific usability problems were clustered into eleven general problem areas and ranked according to severity and the ease with which they could be fixed. The seven most severe problem areas were addressed in more detail in this report, providing information about the general problem, some specific examples, and a high-level recommendation for solving the problem.

The seven most severe and easiest to fix problems are:

1. **Inconsistencies between menus and buttons**
2. **Some language does not correspond with user terminology**
3. **There are buttons that the user may not realize are buttons**
4. **Not all buttons have tooltips**
5. **Some inconsistencies with browsers operating standards**
6. **Undo commands basically unsupported**
7. **Modal interface causes inconsistency in available features**

By investigating these problem areas in more depth and implementing user-centered solutions, Flow designers will be able to make an already well-designed product even easier to use.

---

## RESOURCES

Greenberg, S. (n.d.). *Design Principles and Usability Heuristics*. Accessed February 23, 2005, at [http://pages.cpsc.ucalgary.ca/~saul/hci\\_topics/pdf\\_files/heuristics.pdf](http://pages.cpsc.ucalgary.ca/~saul/hci_topics/pdf_files/heuristics.pdf)

Nielsen, J. (1993). "Usability Heuristics." In *Usability Engineering*. San Diego, CA: Academic Press.

Nielsen, J. (n.d.). *How to Conduct a Heuristic Evaluation*. Accessed February 9, 2005, at [http://www.useit.com/papers/heuristic/heuristic\\_evaluation.html](http://www.useit.com/papers/heuristic/heuristic_evaluation.html)

Nielsen, J. (n.d.) *Severity Rankings for Usability Problems*. Access February 22, 2005, from <http://www.useit.com/papers/heuristic/severityrating.html>

Nielsen, J. (n.d.). *Ten Usability Heuristics*. Accessed February 9, 2005, at [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)

Olson, J. (2005, February 10). *Quick Methods: Checklists, Heuristic Evaluation, Cognitive Walkthrough*. Slides presented in a lecture at University of Michigan School of Information, Ann Arbor, MI.

---

## APPENDIX A: DETAILED HEURISTIC LIST

### 1. Aesthetic and minimalist design

- ❖ Brevity: Keep information to the bare minimum.
- ❖ Categorize: increase readability by categorizing repetitive information into relevant sections.
- ❖ White space: Add breathing room to the information displayed.
- ❖ Color: User's eye focuses on a particular area of the screen.

### 2. Match between system and real world

- ❖ Screen representation matches non-computer.
- ❖ Metaphors from the real world.
- ❖ Familiar user's conceptual model.

### 3. Recognition rather than recall

- ❖ Allow access to operations from other apps.
- ❖ Make the repertoire of available actions salient.
- ❖ Show icons and other visual indicators.

### 4. Consistency and standards

- ❖ Show similar information at same place on each screen.
- ❖ Conform to platform interface conventions.

### 5. Visibility of system status

- ❖ Feedback: show what input has been received.
- ❖ Direct manipulation: visible objects, visible results.
- ❖ Identity cues system response vs. user's goals.

### 6. Flexibility and efficiency of use

- ❖ User tailorability to speed up frequent actions.
- ❖ Shortcuts: Accelerators to speed up dialogue.
- ❖ User control: Allow user to initiate / control actions.

### 7. User control and freedom

- ❖ Modeless: Allow users to control actions.
- ❖ Ability to re-order tasks.