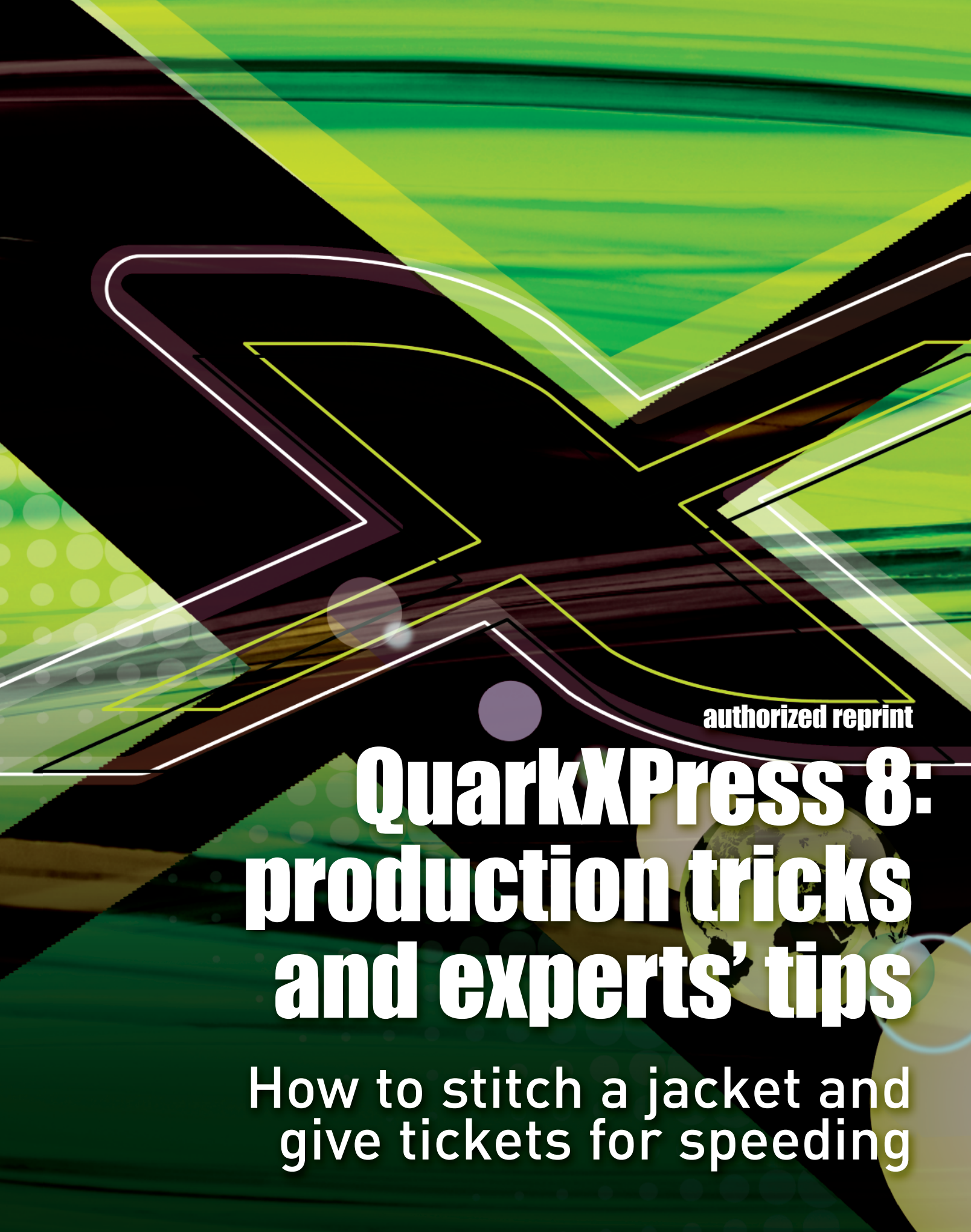


authorized reprint

QuarkXPress 8: production tricks and experts' tips

How to stitch a jacket and give tickets for speeding

Chapter 19: How to stitch a jacket and give tickets for speeding

BY JOE ROOT AND THOMAS ALLEN

So what the heck is the deal with this job jackets thing? Well, to make a long story short, job jackets is a way for us to communicate what we really want to have happen to our jobs as they get built and when they go to press. It's no different than the job jackets that you've used for years — except that it is an electronic representation rather than a big plastic envelope.

Well, that seems simple enough, but you may now be compelled to ask, "How does it work?" The devil is always in the details; let's take a look at the feature's background and then demonstrate some important job jackets functions that can save some serious time in our workflows.

This chapter will help you to understand job jackets, and what is possible. This is a first step so I want to go back to the very beginning. If you're like me, you learn technologies by actually playing with them. I think you'll find this chapter a place where you can get in to learn and master the feature set. At least, that's my intent.

Introduction to Quark Job Jackets®

It would be improper of me to open this article without a little background on job jackets. As I said, I describe this feature differently because I often describe things with respect to what I can see them doing and how I can use them. So, here's my take: I perceive job jackets as functioning on three distinct levels.

CONTAINER

The first level is the container. Job jackets can contain different categories of objects.

- **Style** — QuarkXPress style resources such as style sheets, colors, H&Js, dashes and stripes, and lists. Use this capability to maintain consistency in design or as a repository for commonly used resources.
- **Specifications** — specifications, parameters, contacts, and descriptions come next. You can create layout and output specifications, output styles and setups, contact information, and different types of job descriptions in job jackets. The reason that I conceptually group these settings together is because they are all independent objects that live in job jackets that will later get applied to an actual job.
- **Rules** — special specifications beyond what we can establish in a standard layout or output specs. We can create rules to check for all kinds of things. You will see an example later in this chapter.
- **Tickets** — very special objects in job jackets that are essentially combinations of rules, specifications, and

Job jacket

A job jacket is a container for an entire job that may consist of multiple, distinct pieces. A job jacket's file might represent an entire publication with multiple sections or an advertising campaign with different pieces. A job jacket holds:

- Global, generic data such as contact information, job descriptions, or rules.
- Any number of job tickets; either job ticket templates or active tickets.
- Any number of layout specifications that define particular layouts within the jobs (these can be print, web, or interactive layouts).
- Any number of output specifications that define parameters for successful output.
- Any number of resources, such as style sheets or colors.

style resources that are used to create or *evaluate* an actual job, or more specifically, a real QuarkXPress layout. Tickets are templates, or the driving force, of real jobs. You will see this in action in this chapter.

COLLABORATION

The next level is collaboration. Collaboration can happen in a couple of different ways.

- **Planning** — multiple job planners in an organization could add information to job jackets. On many levels, this could simplify the job-administration process.
- **Shared resources** — multiple people can pull resources, such as style sheets, from job jackets. If they are all using the same resource from the same job jacket and we make a change to that resource, it could change for all the people using that resource. So let's say I create a color called Quark Green, or Pantone 369 C, and I load it into my job jacket. Four other designers are using that job jacket for their projects. After some time I realize that Quark Green is actually Pantone 368. All I need to do is change that color at the job jackets level and it will change for the other four designers dynamically.

In short, job jackets is a way for us to communicate what we really want to have happen to our jobs as they get built and when they go to press.

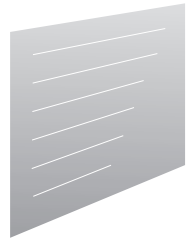
EVALUATION

The third level is evaluation. Setting all these rules and specifications is really nice, but how do we make sure that we are staying consistent with the rules we've set? Quark job jackets comes with an evaluation module that allows you to check your work and make sure that it is built in the correct manner. You'll also see an example of this function in this chapter.

Job ticket

The job ticket contains:

- The output specifications applicable to that job.
- Layout definitions for each layout within the project.
- Layout specifications applied to each layout definition.
- Specific resources for the job.



Jumping in

With the basic intro out of the way, let's actually get our feet wet and play around some. We are going to automate the creation of a QuarkXPress layout with job jackets. So what are the steps for proceeding? It's a four-step process:

1. Build a job jackets file.
2. Create a job ticket.
3. Use that ticket to automatically generate a QuarkXPress layout.
4. Create our job and evaluate that job for compliance with the ticket and job jackets specifications (that we defined in steps 1 and 2).

Part 1: Stitching your first job jacket

So let's get started with the job jackets manager; we need to create an actual jacket. Job jacket functions can save some serious time in our workflow.

1. Launch QuarkXPress 8, but don't open any files.
2. Go to **UTILITIES ➲ JOB JACKETS MANAGER.** You should see that there is at least one default job jacket in the current view, but we will create our own.
3. Click the **NEW JOB JACKET BUTTON** (the first button from the left represented by a sparkling folder in the job jackets dialogue box).
4. Name the job jacket *Test Jacket*. (Optionally change the default save location.)
5. Click **SAVE**. You should have something that looks like figure 1. If you've been here before, you may have more than just the default job jacket, as I do.

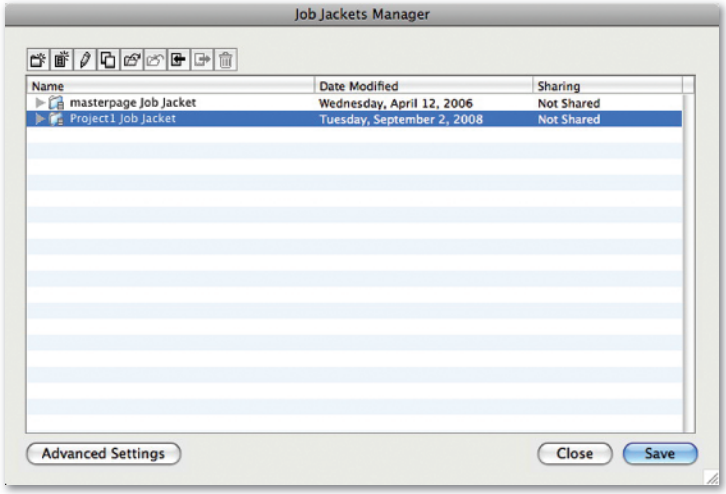


Figure 1
The bulk of your interaction with job jackets will be through the job jackets manager dialogue box.

BASIC MODE

This view of the Job jackets manager in figure 1 is basic settings mode. There is an advanced settings mode as well, but we are going to try to keep it simple in the beginning. Now that we have a job jacket, let's add some resources for later use.

step 1 Click on the title of the new jacket that you have just created.

step 2 Click on the **EDIT BUTTON** (the third button from the left, represented by the pencil icon).

step 3 Click the **DISCLOSURE ARROW** next to **SETTINGS**. You should now see four tabs: **TICKETS**, **STYLE SETTINGS**, **CONTACTS**, and **LAYOUT SPECIFICATIONS**.

step 4 Click on the **STYLE SETTINGS TAB**.

THE STYLE SETTINGS TAB

Quark job jackets hold four major categories of objects: *rules*, *specifications*, *tickets*, and QuarkXPress *resources* such as style sheets, colors, and even H&Js. You can use job jackets to store style resources to maintain consistency or to act as a repository for commonly used resources.

APPENDING STYLE RESOURCES

Here's how I append my style resources. You can follow along with a document of your own, if you wish.

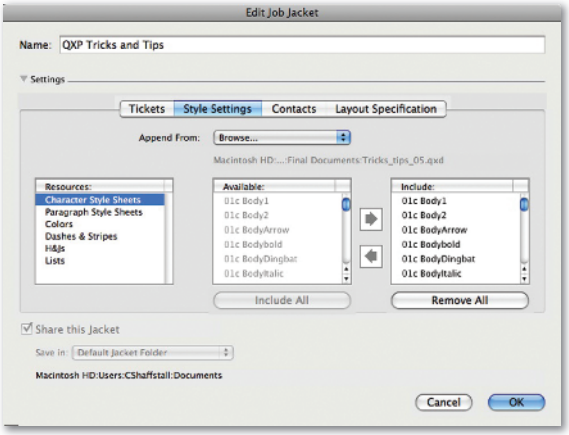


Figure 2
You don't have to start from scratch. Job jackets enables you to append resources such as style sheets from existing projects, like I've done here.

step 1 From the **APPLICATION DROP-DOWN MENU**, choose **BROWSE**.

step 2 Navigate to the QuarkXPress project from where you wish to collect style resources (such as style sheets).

step 3 Click **OPEN**. After locating the file, QuarkXPress displays the different resources available for import.

step 4 Click on **CHARACTER STYLE SHEETS** in the left-most pane of the dialogue box, labeled **RESOURCES**. When you do that, a list of available style sheets from the file you appended will appear in the available column. (Be sure to choose a file that you know has resources. It's not a great example if you only have normal.)

step 5 You may import a single style sheet or all style sheets at one time. For just one style sheet, double click on the title of the style sheet in the **AVAILABLE PANE**. It will now appear in the **INCLUDE PANE**. You can see my dialogue box in figure 2. After I imported that one style sheet, a color was automatically imported as well. I have two Pantone colors in these style sheets, so they are automatically included as well.

step 6 Repeat this process to load additional style sheets, colors, dashes and stripes, H&Js, and even lists to your job jacket.

step 7 Click **OK**.

CONTACTS

Contacts are important for jobs. I remember sending out jobs and the most important contact was the person to call at 2 AM when something goes wrong. You add contacts to job jackets as well — as many as you want — but for my project, I am going to add just one for now: myself. To add a contact to your job jacket, follow along:

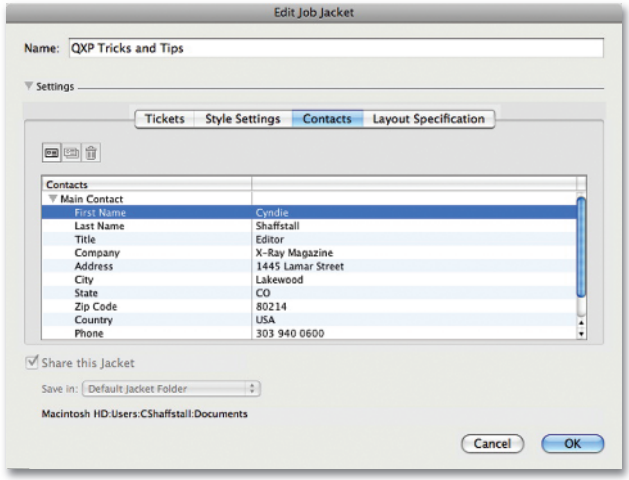


Figure 3
As with any traditional job, you may have several different contacts; perhaps the CSR, the prepress manager, the shop owner, or your buddy in the press room.

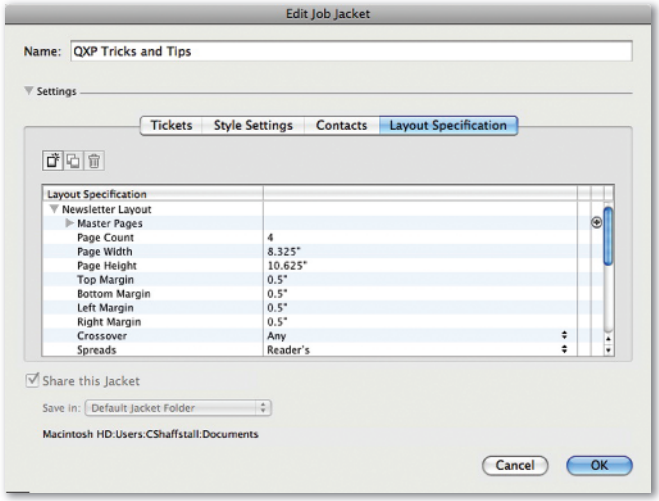
1. Click the **CONTACT TAB**.
2. Click the **NEW ITEM BUTTON** (the left-most button). Doing this will create an entry in the contacts pane titled *contact*.
3. Click on the title **CONTACT** so that you may edit it and choose a title for this contact person. I have chosen **MAIN CONTACT**.
4. Click the **DISCLOSURE ARROW** to the left of the contact label. A series of editable fields becomes available in the right side of the **CONTACTS PANE**.
5. Complete these fields with your preferred information, as I have done in figure 3.
6. Click **OK**.

That's all there is to adding contacts. Create more contacts, if you wish by clicking the new contacts button again and entering information for others that may need to be involved in this project.

THE LAYOUT SPECIFICATION TAB

A layout specification essentially sets your layout properties, just as you do when you start a new QuarkXPress project. For example, if you need to create a job, and that job must be 8.325˜ wide × 10.625˜ tall, you can predefine that parameter here. To create a layout specification, follow these steps:

Figure 4
The layout specifications enable you to predefine the intended job size, colors, page number, and so on. Much like the new project dialogue box, but more.



1. Click on the **LAYOUT SPECIFICATION TAB.**
2. Click on the **NEW ITEM BUTTON**. This will create a new layout specification into the dialogue window below.
3. Click on the default title of the layout specification. The box will change so that you can edit the name of the layout specification. Be sure to use something descriptive since you may have many layouts in a project (such as *postcard*).
4. Click the **DISCLOSURE ARROW** next to the title of your layout. This will list all of the parameters that you can set as specification for this layout. Notice that many of these specifications are what you would find in a **NEW PROJECT DIALOGUE BOX** if you were creating this project manually.
5. Set **PAGE COUNT** to **4**. This means that our layout will contain four empty pages.
6. Set **PAGE WIDTH** to **8.325˜**.

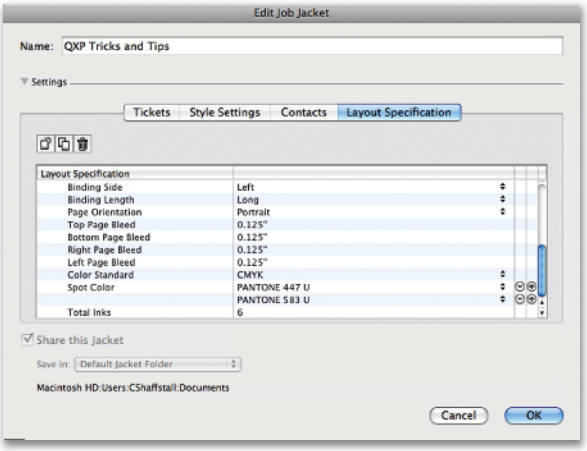


Figure 5
While my project is CMYK, I also have two spot colors. I have identified them here. Now I don't have to worry about one of the project contributors — or me — inadvertently using the wrong color.

Step 7 Set **PAGE HEIGHT** to **10.625˜**.

Step 8 Scroll through the rest of the parameters to become familiar with the options available and define any that you choose. See figure 4. *Optional*: Scroll to the bottom of the layout specifications list to spot inks. If you want to add more than one spot color, click the **PLUS SIGN** next to the far right of this line and define a second color. Check out figure 5.

Step 9 Click **OK** to return to the main **JOB JACKETS MANAGER** DIALOGUE BOX.

Step 10 Click **SAVE**.

Review

So what did we just do? We have created a job jacket and added resources. In this example, I added style sheets, two colors, one contact, and one layout specification. What do we do now? Let's add more stuff to the jacket.

Advanced mode

At the bottom of the **JOB JACKETS MANAGER** DIALOGUE BOX click the **ADVANCED SETTINGS** BUTTON. See figure 6.

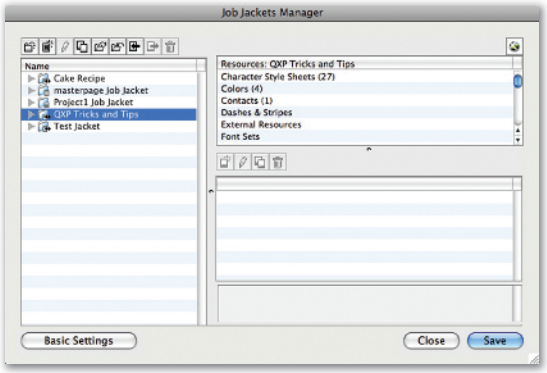


Figure 6
In the advanced settings mode, you define rules and rule sets.

RULES AND RULES SETS

In figure 6, we see that we already have some specifications in this jacket that we added while in basic mode. Another category of resources that can be contained by job jackets is rules.

Why do we want rules? Imagine a project where people accidentally use RGB images when they are supposed to use CMYK. Imagine a day when people prefer the bold styling button to a bold font. Imagine a day when people use spot colors in a four-color job. WHAT?! You don't have to imagine it? It happens all the time?…

With job jackets, we can look inside our job and learn about things we've done that won't output with our preferred printer, or that is not allowed by our customer. We can learn about these things as we do them — long before it becomes a critical error on a press that's going to cost us time and money.

DEFINING A RULE

You have complete control over what you will allow in a job. Using rules, you can remind yourself, or set up parameters for others. Let's start by creating our first rule.

Step 1 With your job jacket selected, choose **RULES** from the scrolling list in the top-right pane of the **JOB JACKETS MANAGER** DIALOGUE BOX.

Step 2 Click the **NEW ITEM** BUTTON between the two right-hand panes (hover over these buttons for tool tips to describe them).

3 In the resulting dialogue box, and with the default name still selected, type to name the rule **NO BOLD STYLING**.

4 From the **SUBJECT DROP DOWN**, choose **TEXT CHARACTERS**.

5 In the **CONDITIONS LIST**, check the box next to **TEXT STYLE**.

6 Click **NEXT**.

7 From the **OPERATOR POP-UP LIST**, choose **IS**.

8 From the **STYLE TYPE POP-UP LIST**, choose **BOLD**.

9 Click **NEXT**.

10 In the **DESCRIPTION FIELD**, type a message that will be displayed if the specification is not met.

11 From the **POLICY DROP-DOWN MENU**, choose your policy toward bold styled text. It can be **NOTED**, **NOT RECOMMENDED**, or **PROHIBITED**. Choose **PROHIBITED**.

12 In the **INSTRUCTIONS FIELD**, type a message to the designer that tells them what to do should they commit this faux pas.

13 Click **FINISH**.

14 Click **SAVE**.

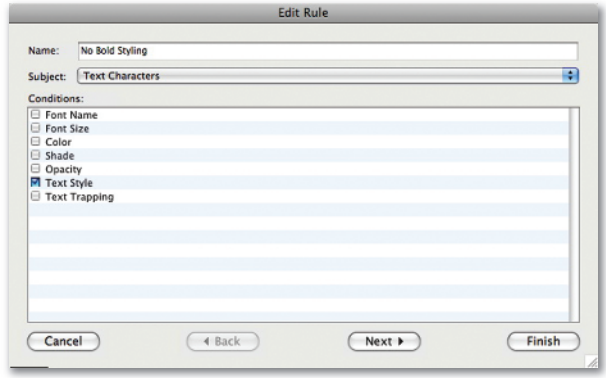


Figure 7
Rules are categorized as shown above. Select a category for which you wish to assign a rule.

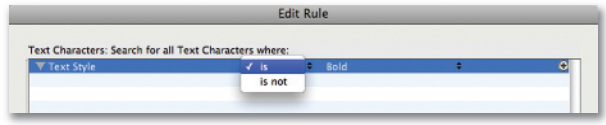


Figure 8
Click the double opposing arrows to display the drop-down list of options for any rule. Here we have defined a rule that if text style **IS** bold, then provide an alert.

Take this opportunity to create more rules of your own and discover how the conditions and operators work for each of the different type of subjects.

RULE SETS

Rule sets are a way to organize rules that you have created. Using rule sets, I can isolate specific rules for specific problems or even special jobs.

Now that we have a rule for bold listed, I will add another related rule in a slightly different manner. I selected the No Bold styling rule in the list and clicked the duplicate button just above it. I want this rule to be exactly the same as the previous rule except that it is named appropriately and contains instructions for italic.

Our job isn't done. You cannot use rules by themselves. You must add them to rule sets. Right above rules in the advanced settings mode of the job jackets manager we see the rule sets resource.

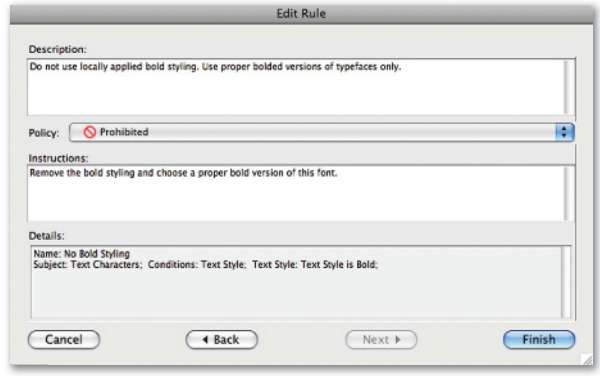


Figure 9
You have plenty of space to describe the rule, choose the policy about the rule (from a drop-down menu), and then describe the steps to take to rectify the disallowed instance.

Rule sets are a way to organize different rules. For example, I could have 30 rules listed here but not all of them will apply to every job that uses this jacket. So using rule sets, I can isolate specific rules for specific problems or even special jobs. Continuing with my example, I have created a new rule set.

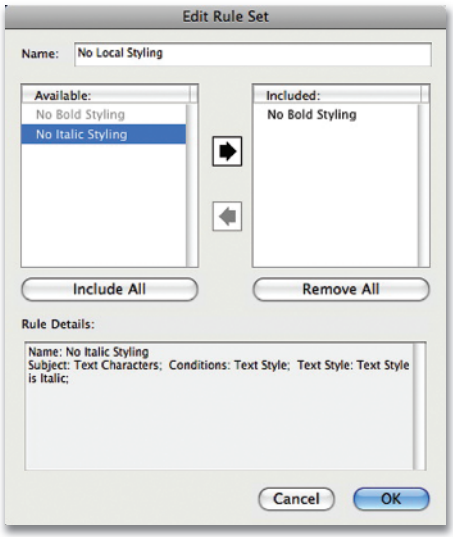


Figure 10
Rule sets are containers for rules. You might wish to create lots of rules, but not all apply to every job. By creating a rule set you can include specific rules and omit others. A rule may be included in as many different sets as you choose.

DEFINING A RULE SET

If you'd like to follow along and create a rule set, these steps will help you:

1. Continuing with your job jackets in **ADVANCED SETTINGS** MODE, select **RULE SETS** from the scrolling **RESOURCES** PANE.
2. Click the **NEW ITEM BUTTON** between the two panes on the right side of the dialogue box. It's the same button you used to create a rule.
3. Type a name for the rule set. (In my example I have named it *No local Styling*.)
4. In the **AVAILABLE** PANE, double click the title of each rule that you want to add to this rule set. When you double click the title it is added to the **INCLUDED** PANE. If you prefer, click **INCLUDE ALL**. If you want to review the content of a rule, click once and the details are displayed in the **RULE DETAILS** PANE.
5. Click **OK**.
6. Click **SAVE**.

I have successfully created a new rule set and I am on my way. I assume you're right here with me.

Output specifications

Before we move forward, let's talk about output specifications. This is the category of specifications that can validate whether or not you have set up your job for output the right way. You don't want to go through your whole job thinking this is going to be printed locally with RGB images if the final output is actually going to be CMYK.

To create output specifications of your own, follow along:

1. While your job jacket is selected and in **ADVANCED** MODE, select **OUTPUT SPECIFICATIONS** in the **RESOURCES** PANE.
2. Click the **NEW ITEM BUTTON**.
3. Click the default name and provide a new name or your output specification.
4. Click **SAVE**.

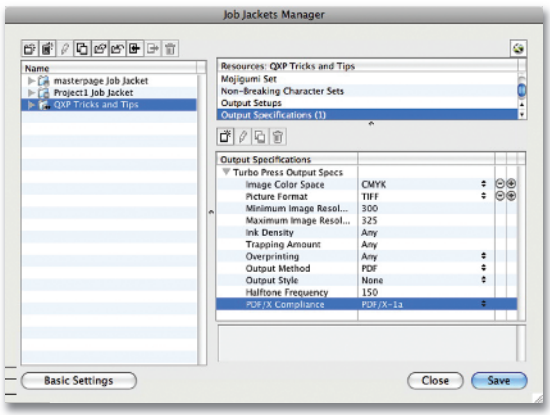


Figure 11
Output specifications let you pass forward information that you may have about the output process. This could be augmented by information from your printer, or may just be what you know thus far about this project.

In figure 11 you can see that I have clicked the **NEW ITEM BUTTON** and I have named the output specification *Turbo Press Output Specs*. For my example, I set a number of the parameters. Below is a description of those settings. Now that you know how to set parameters, feel free to experiment and set some of your own.

- **Image Color Space**: CMYK – Only use CMYK images.
- **Picture Format**: TIFF (To add more than one image format; use the **PLUS SIGN** to the far right. It will drop down another menu item where you can specify additional formats.)
- **Minimum Image Resolution**: 300 – Don't use imagery that is set lower than 300 DPI.

- **Maximum Image Resolution**: 325 – Don't use imagery that is set higher than 325 DPI.
- **Output Method**: Print – This is very important. If this is set to print, you will not be able to use this output specification with a web or interactive layout (this is good since both of these electronic formats would use images at far less resolution than 300 _ 325 DPI).
- **PDF/X Compliance**: PDF/X-1a – This job should be output as PDF/X-1a.

This is enough for now. We have a good start by adding just a few specifications to the container, or jacket. Now we need to associate these resources with a job. This is where job tickets come in.

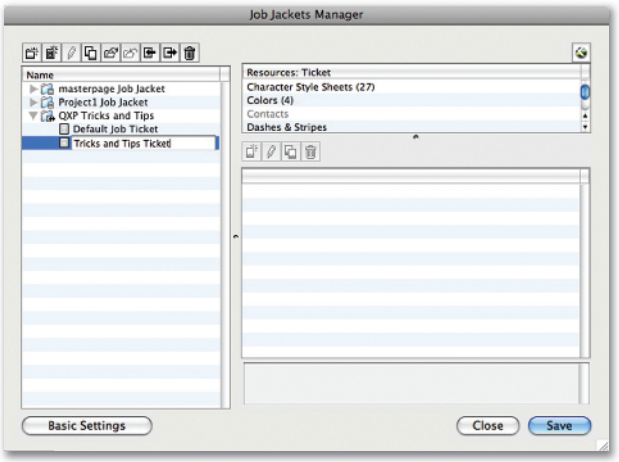
Tickets are the law

Job tickets are very special resources that are also stored in the job jacket file. Essentially, tickets are different combinations of jacket resources that are brought together in one place. You choose a layout specification (if you have one), an output specification (if you have one), rule sets (if you have them), and collect them into a ticket. Why?

The ticket is the combination of specifications, rules, and resources for a specific job. You could have tons of stuff in one jacket, but not all of it applies to a specific job. The ticket is what creates the standard for the specific job, or better yet, it is the template for how you set up, build, and output that specific job.

Let's now apply some of the resources we have put into our job jacket to a ticket that will be used for a specific job. To qualify that just a little more: if we don't apply these resources to the ticket, they won't be immediately available from the ticket when we use the ticket for a job. However, that doesn't mean that we can't get to them later, it just means that they aren't available automatically when we start the project. Take a look at figure 12.

Figure 12
Now that you have create a job jacket, you need a job ticket. Click the new ticket button to create the ticket, then click the title to rename it.



Adding a new ticket

To create a job ticket for your project, follow these steps:

1. In the **NAME** PANE of the **JOB JACKETS MANAGER** DIALOGUE BOX, select your job jacket. Be sure that you are in the **ADVANCED SETTINGS** MODE.
2. Click the **DISCLOSURE** ARROW to the left of your job jacket's title. This enables you to view what tickets are available inside the jacket. The only ticket available here is the default job ticket. You can work with that ticket, but let's create a new ticket instead.
3. Click the **NEW TICKET TEMPLATE** BUTTON (the second button from the left indicated by a sparkling ticket icon).
4. Click the default ticket name to type a new name.
5. Click **SAVE**.

ADDING RESOURCES TO A TICKET IN ADVANCED SETTINGS MODE

My example is shown in figure 13. I have my ticket selected in the name pane. In the resource pane I have character style sheets selected. In the bottom-right pane, we can see my extensive list of style sheets.

Notice that the status of the style sheets are currently identified as In job jacket. That means that we have added that resource to the job jacket container but we haven't applied it to this ticket. As a matter of fact, all resources that we have added thus far exist only in the job jacket. To add these to the job ticket, do the following:

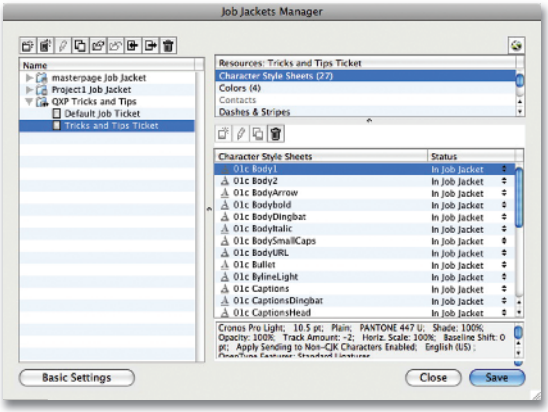


Figure 13

At this point, the style sheets are contained within the job jacket, but not the ticket. Click the pop-up to add selected style sheets to this ticket, or add them all.

1. Select **CHARACTER STYLE SHEETS** from the scrolling list in the **RESOURCES** PANE.
2. Click the **STATUS** POP-UP LIST in the lower pane and set it to **IN TICKET** for each character style sheet that you wish to add to this ticket. Only those resources that you set as **IN TICKET** will become resources for the projects created from this ticket.
3. Continue through the scrolling list of resources and change the status to **IN TICKET** for each resource you wish to add. Be sure that your ticket is selected in the name pane as you add resources.
4. Click **SAVE**.

Adding layouts to tickets

The resources we have added to the ticket so far will define a future job (QuarkXPress layout). We would want to do this for two important reasons. The specifications will help us to make sure we are building our job correctly and they will automate the creation of the layout.

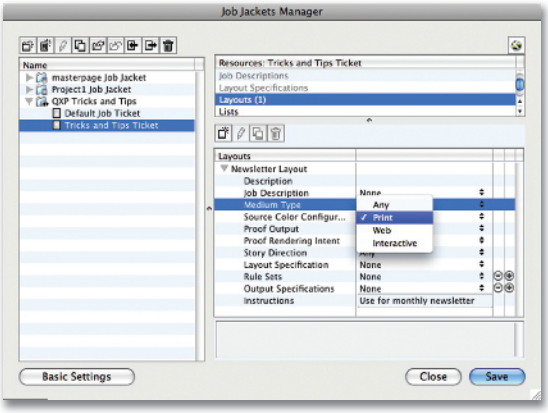


Figure 14

Adding a layout to your ticket will ensure that the proper type of layout is created when this ticket is used.

To add a layout to a ticket, follow these steps:

1. Select your ticket in the **NAME** PANE of the **JOB JACKETS MANAGER** DIALOGUE BOX.
2. Select **LAYOUTS** in the **RESOURCES** PANE.
3. Click the **NEW ITEM** BUTTON to add a new layout.
4. Rename the layout **MY LAYOUT** by clicking on the title.
5. Click the **DISCLOSURE** ARROW to the left of your new layout.
6. Click **SAVE**.

Now let's walk through the settings options. Here is a quick explanation and steps for continuing with our example.

- **Description**: A description of the layout.
- **Job description**: When job descriptions are added to a jacket, they can be selected from here. This is done in the same manner as adding contacts or layout specifications.

- **Medium type**: Very important, and in my example I will choose print.
- Source color, proof output, and rendering intent are all color-management settings that we've not yet explored. Let's skip this for now.
- **Layout specifications**: If you have followed along, you will have a postcard layout specification listed here. Click the double arrow and select it.
- **Rule sets**: Click the double arrow and select the rule set you created, if you have been following the steps throughout this article.
- **Output specifications**: Again, if you have been following the steps, click the double arrow and select the output specification that you created.
- **Instructions**: Add any additional notes here.

Reviewing the ticket

We've taken some of the resources from the jacket and added them to the ticket so they can be loaded into a future layout. Then we added combinations of other resources that will generate that future layout.

Part 2: Building a layout from a ticket

Now that we have a ticket that contains a specification for a layout, all we need to do now is test drive our new job jacket.

1. Click the **SAVE BUTTON** in the **JOB JACKETS MANAGER** DIALOGUE BOX.
2. Click the **CLOSE BUTTON** to dismiss the dialogue box.
3. Go to **FILE ➲ NEW ➲ PROJECT FROM TICKET.** After doing that, a mini jacket viewer pops up. Since we were just working on a jacket and ticket, the ticket we want is already open. If our jacket isn't listed here, browse to and select the ticket you have created. In figure 15 you can see that I have one jacket open, and my ticket is displayed.
4. Select the ticket name that you created.
5. Uncheck the **SHARE JACKET CHECK BOX**. (We'll save this for a future discussion.)
6. Click **SELECT**.

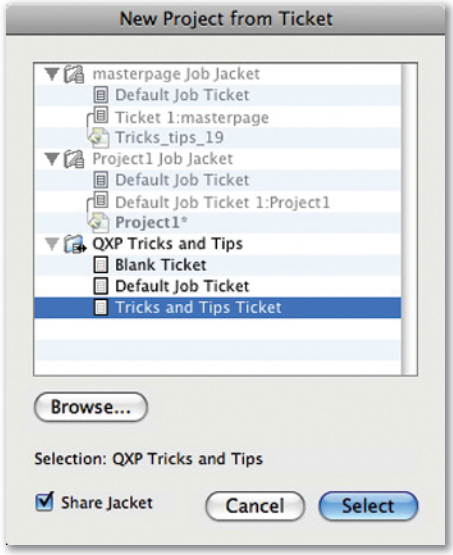


Figure 15
Go to the file menu and instead of choosing new, choose project from ticket. Once you receive this dialogue box, choose the ticket.

So let's take a look at what has happened. When I click the select button shown in figure 15, QuarkXPress 8 read the job ticket that I created, made note of all of my resources, and simply built my layout exactly as I specified.

The layout is named newsletter layout, it is four pages long, and it is even 8.325˜ × 10.625" just as I specified. Further, it automatically brought in the style sheets that I appended and the two Pantone colors. What a start to this job. The entire layout structure was built automatically from the ticket and built the right way from the beginning. Now there's no chance that I will create a three-page job when the specifications call for four, and I won't run the risk of making it 8.5˜ × 11˜.

I created my layout using the job ticket and started building a page. intentionally using an RGB image and bold and italic text on the page. In the rules I created I specified that I am not allowed to have RGB images or bold or italicized text, but I did it anyway…

Job Definition Format; what is it?

Job Definition Format (JDF) is a standard that is managed by the International Cooperation for the Integration of Processes in Prepress, Press, and Postpress Organization (CIP4). The concept of JDF is to standardize the exchange of information between different systems and applications within the graphic arts industry. The JDF specification has primarily focused on processes that begin at prepress and carry through post-press, though several CIP4-managed working groups are focusing on expanding the specification to account for all processes in the industry, including content-creation processes.

Part of the JDF specification deals with Job Messaging Format (JMF); the method that JDF devices use to communicate with each other. The goal is to allow any device in a workflow to communicate with and command any other device. There are already JDF devices that can talk to each other — a device, for instance, can communicate that it is capable of producing a job as specified in a JDF specification; another device can communicate that it can't produce the job because it is out of necessary supplies.

QuarkXPress was the first application to bring the concept of JDF into the content-creation space. While other applications do support JDF, that support is provided only at the end of the process — so if users encounter a problem, they'll need to switch applications, fix problems, and repeat the process until the file is correct. With QuarkXPress, creating a new project from a job ticket means the specifications from your output provider — their particular rules and requirements for effectively producing the job — can be present and affect a layout from the moment it's created. When you're ready to output, there's little need to run a separate preflight or check; the evaluation has been happening all along.

When you output your QuarkXPress layout, you have the option to also output a JDF file. This file makes the information from the job jackets file available so that it can be read and processed by other JDF-enabled systems. The option to include the JDF file is available with any type of output, not just JDF, and can be stored as part of an output style. Since job jackets technology is compliant with Quark's XTensions technology, output providers and vendors of JDF-enabled technology can even write custom information into their job jackets files. Such information travels with the job and is included in the JDF output.

Just as the JDF specification is evolving, so will QuarkXPress in its adoption of JDF. QuarkXPress 7 introduced job jackets, but it's just that — the introduction of this technology and JDF into QuarkXPress workflows.

Part 3: Evaluation and electric shock

So you may ask, "What good are rules if I can just break them at will?" Brilliant question!

The makers of job jackets did not want to transfer electric shock through the mouse if you did something wrong, they just wanted to tell you when you've done something outside the intent of the job — leaving the final decisions entirely up to you. The idea of rules and specifications isn't to stop you from working; it is to flag potential problems so that you don't have costly errors later at output time. So how do we flag problems? Let's take a look at evaluation, and you can follow along with these steps:

1. Go to **FILE ▸ JOB JACKETS ▸ EVALUATE LAYOUT**.
2. In the **LAYOUT EVALUATION** DIALOGUE BOX, click the **EVALUATE** BUTTON.
3. After the evaluation, click on the listed rules and specifications to see what mistakes were made.
4. Click the **SHOW CASE FIRST**, **SHOW CASE PREVIOUS**, **SHOW CASE NEXT**, and **SHOW CASE LAST** BUTTONS to go to problem objects.

My evaluation is shown in figure 16. As planned, because I have used the bold styling and placed an RGB image in my project, the evaluation process displays a message that I have failed. We knew this would happen because my document contains bold text, italic text, and even an RGB image on the page. I defined the specifications to allow the use of TIFF images but the balloons image is a JPEG. If I select one of the listed issues in the dialogue, I can click the show cases buttons to show me where I messed up. Not only do I get to see what's wrong, QuarkXPress 8 will even show me what needs to be fixed. Clicking done will close the layout evaluation dialogue.

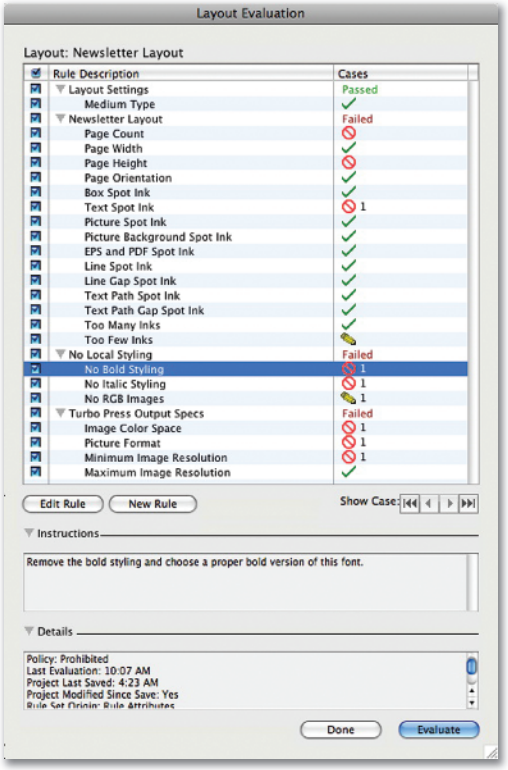


Figure 16
When I run the evaluate process, I receive alerts for all areas where I have violated the rules of the job ticket.

Sample job jackets workflows

Let's look at some sample workflows and explore some features that might hold important keys to improving your workflow with job jackets technology. If you'd like to take a look at these or any of the examples in this article, you can find the job jackets file at http://www.xraymag.com/downloads/jobjackets.zip.

JOB JACKETS CASE #1: I HAVE A BORING JOB

You're not collaborating with anyone. You work with one vendor all the time. You're a perfect candidate for the default job jackets file. "What's that?" you ask.

Every project opened in QuarkXPress 8 (and 7) has an embedded job jackets structure. This structure comes from the default job jackets file that is located in the QuarkXPress preferences folder. To add resources to the default job jackets file you have to open it in the job jackets manager, just like you would any other job jackets file.

Using the skills you've just developed, edit your default job jackets file, add a layout specification for the newsletter, create some rules, and add that handful of style sheets. Now apply all of those settings into a layout definition in the default job ticket and save. Use the job ticket to create a new project and gain an immediate advantage on the production cycle.

JOB JACKETS CASE #2: I WANT TO BE ALONE WITH MY DESIGN

What if you're in a workgroup and you want the benefits of job jackets technology? You want your layout to be automatically color managed to your output provider's press, you want the power of auto-evaluating layouts, you want to get all the specifications and resources from the moment you create a layout; yet, you're not really into that collaboration hoo-ha.

The job jackets file in this workflow is still the point where all projects originate. When our designer creates a new project from ticket, she's presented with the selection dialogue box. All she has to do is select the appropriate job ticket template and uncheck the share jacket check box in the selection dialog. Presto! She has all the rules, specifications, and resources, and they are all embedded in her project. There's no collaboration involved, but she likes it that way.

In this example, the job jackets file can be updated at any time without affecting anyone else in the workgroup. It seems odd, sure, but this might just be the workflow that you're looking for.

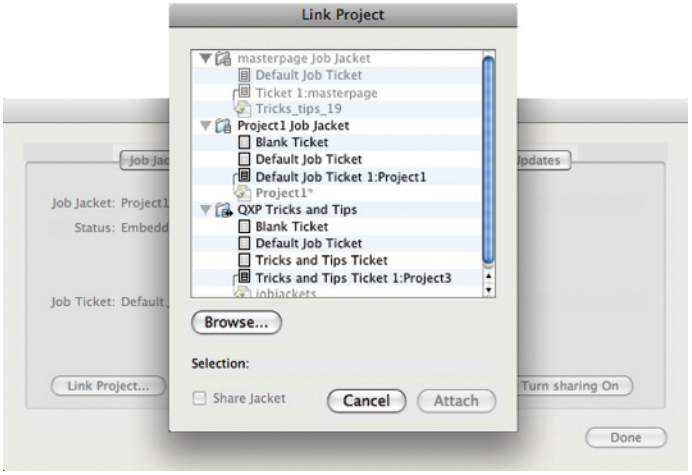


Figure 17
Adding a layout to your ticket will ensure that the proper type of layout is created when this ticket is used.

JOB JACKETS CASE #3: I LIKE YOU; YOU LIKE ME

You're a small design group. You and your fellow designers are all working independently on concepts for the new job but you need some serious brainstorming and mind melding. Maybe you couldn't care less about an output specification, and maybe you think JDF stands for Jane's Double Fudge, but you can still use job jackets structures to share your creativity.

Start by sharing your job jackets file as a collaboration point. Save it in a location where your colleagues can access it. Decide what resources you want to share. If you want their stuff you'll have to get it from the job jackets file just like they got yours. From then on, a change to one of the resources will be made in everyone's project. Just don't forget that in this wildly collaborative place, a compliment goes a long way.

JOB JACKETS CASE #4: X-RAY MAGAZINE

X-Ray Magazine works with an output provider who can collaborate in a standard job jackets file to include their output specifications, complete with output styles and custom color-management settings. There's a definite benefit to managing resource definitions in the job jackets file, for sure. If a style sheet needs tweaking, it'll update in all the files automatically. Like most magazines, *X-Ray Magazine* works with templates so while page sizes, for instance, will not be modified, the job jackets file will be propagated when a template is opened and altered page size will produce rule violations.

JOB JACKETS CASE #5: THAT OLD, FAMILIAR AGENCY FEELING

When that phone call comes through that your agency is awarded the job, start where you normally do. Perhaps the account executive creates the job numbers for billing purposes and involves the traffic manager.

You are involved immediately and create a job jackets file on the file server. You enlist the traffic manager to add all the job descriptions complete with the assigned job numbers. It's one less thing you have to do. You solicit bids from different output providers, entering all the pertinent contact information directly into the job jackets file. Some of the output providers are experienced in job jackets creation and offer their own job jackets containing their output specifications and rules. So you take advantage of that, and drag and drop the resources from the provider's job jackets into your job jackets file. Brilliant. See? You get it!

Review

This chapter really only provides a narrow illustration of the power of job jackets. Sharing job jackets in a workflow with others can add even more power to this feature set. It will bring a continuity and security to even the most complex projects, but providing plenty of latitude for individual control over project components.