Part I

Check-Mate Essentials

In This Part

- Understanding what Check-Mate is and what it can do for you
- Knowing how to start using Check-Mate right away

In this part, you find out about the basics of Check-Mate, and I briefly introduce you to some of the other UGS applications that connect to Check-Mate. Then I discuss *checks* and *profiles* and tell you how to start using each of them right away.

Once you start using Check-Mate, you're going to spend a whole lot less time going back and fixing all those little problems that somehow slip through the cracks today — because with Check-Mate, you find those little things *before* they become big problems later.

So What Exactly Is Check-Mate?

At its simplest, Check-Mate is an application inside NX that helps you:

- Choose a set of model validation checks.
- Apply those validation checks to your design.
- Report the results of the checking.
- Act on the results.

The checking engine

Every NX user can reap immediate benefit from using Check-Mate, with more than 200 configurable validation checks for drafting standards, modeling and assemblies best practices, and other model quality standards available with zero configuration effort required.

At its core, Check-Mate takes advantage of Knowledge Fusion, the Knowledge-Based Engineering rules evaluation engine that is a core part of every seat of NX. We recognize that in no way can we fully anticipate everything that all our customers will want to check (we don't know all of your required model attributes or your layering convention, for example) so we built Check-Mate to be very easily extendable. You can use the Knowledge Fusion language and our other Application Programming Interface (API) languages to modify existing checks or create entirely new ones to meet your specific needs.

In addition to giving you the ability to check designs while you interactively use NX, we provide a command-line version of Check-Mate. You can use this tool to validate your parts in batch mode, without starting up an interactive NX session. This is really handy for checking large sets of data converted from other CAD systems, or for running periodic (weekly or even nightly) checks for your entire design group.

Check-Mate can optionally make it harder for users to *avoid* validating their designs. Companies can choose the level of automation and enforcement they want to impose with respect to Check-Mate validation.

The interactive Check-Mate tools and the command line utility (`ug_check_part`) can both be used either in a native operating system environment or within your Teamcenter Engineering data management environment.

The reporting engine

After you run a bunch of checks against a design, you want to see the results, of course. Another really important part of Check-Mate is the reporting capabilities.

For users running NX interactively, Check-Mate shows you immediate feedback on the checking results right inside the Check-Mate dialog, and then lets you start fixing any errors that Check-Mate found.

Many of the people who want to see the results of model quality checking may not be skilled NX users. In addition to the interactive tools, we provide a tool called the *Quality Dashboard* for viewing validation results outside of NX. This standalone Java-based tool is perfect for folks like engineering managers who don't need a seat of NX on their desk. In addition to providing some analysis tools for looking at trends in the checking results, the Quality Dashboard can create beautiful HTML-based reports with color graphs and charts.

The Quality Dashboard can similarly be used either in a native operating system environment or within your Teamcenter Engineering data management environment.

. . . and taking action

Check-Mate also integrates very nicely with Teamcenter Engineering so that Teamcenter workflows can "act" on the results of your checking. For example, when you release a design to manufacturing, you likely want every single part to meet your company data quality standards — and if they *don't* meet the standards, then you don't want to release them. Check-Mate and the Teamcenter validation object can automate this final check and ensure that only high-quality data is being passed downstream in your organization.

Great! Show Me Everything!

Okay . . . Here's the big caveat. Due to the size limitations of this particular book format, I can only talk in detail about the interactive NX part of Check-Mate. I have to leave topics like the Quality Dashboard and the Teamcenter integration for another book.

What I *will* talk about, though, are some great strategies for making Check-Mate such a smoothly integral part of your NX design work that you won't believe you used to work without it.

With that, let's dive in and see how this all works!

What Is a "Check"?

A ***check*** is a small piece of logic that looks for a particular condition within a model. Checks can validate anything from layering conventions to drafting standards to various modeling and assemblies best practices.

Just in case you've never actually used Check-Mate, I walk you through how to run a few out-of-the-box checks against an NX design. This exercise shows how you can use Check-Mate without any configuration whatsoever.

If you'd like to follow along in NX, you can find instructions on the last page of this book for downloading the example part file and all of the example checks and profiles in this book.

1. **Start NX and open your part.**

   For the sake of illustration, I use the part shown in Figure 1-1. Though you may not know from just looking at it, this part contains a few tricky problems, just like all good Check-Mate example parts should.

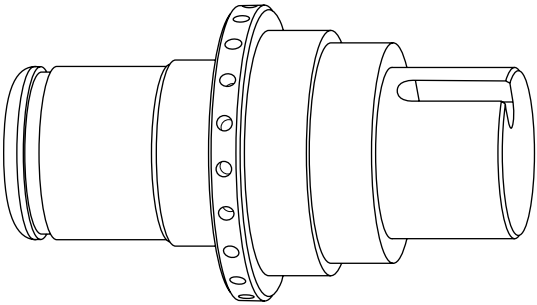

Figure 1-1: Today's beautiful Check-Mate example part.

2. **To start the Check-Mate application, choose the icon labeled Run Tests on the Check-Mate toolbar. (See Figure 1-2.) If you prefer the pull-down menus, choose Analysis➪Check-Mate➪Run Tests.**

   Check-Mate has its own toolbar inside NX. If you can't see it yet, you can turn it on by right-clicking in the toolbar area and choosing Check-Mate from the list of available toolbars.

Figure 1-2: The Run Tests button on the Check-Mate toolbar.

3. **To choose the part(s) you want to check using Check-Mate, make sure that the Parts tab is selected, as shown in Figure 1-3. Select Current Part at the top of this tab to check the current part.**

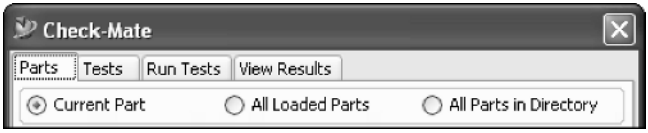

Figure 1-3: Be sure that the Parts tab is selected.

The **All Loaded Parts** option is a good way to check all the parts in a simple assembly — load the entire assembly and choose this option. The **All Parts in Directory** option is a great way to check the results of a translation or verify a set of data received from a supplier or OEM. The folder specified in this third option can be either a folder in the operating system or a folder in Teamcenter Engineering.

4. **To choose the tests you will run, click the Tests tab (see Figure 1-4).**

   Lurking on this tab is a beautiful tree structure containing all of the checks currently available to you.

5. **From the tree, find and select each check you want to run, and click the Add to Selected down arrow button to add it to the Chosen Tests area in the bottom half of the dialog.**

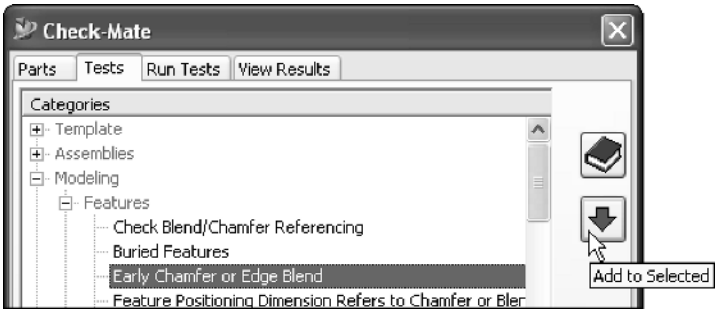

Figure 1-4: The Add to Selected button.

Figure 1-5 shows what the Chosen Tests area might look like after you've chosen a few tests to run and moved each of them down.

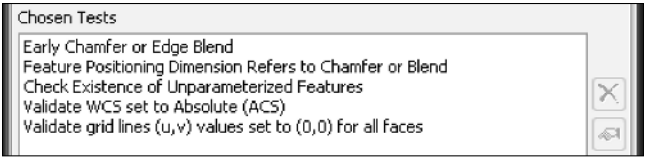

Figure 1-5: The Chosen Tests area, with five tests chosen.

When you have some free time, poke around in this list. You'll see a huge variety of checks available here. They're not just modeling checks, but checks for drawings, assemblies, routing, and other things as well.

6. **To run the selected tests against the selected part(s), click the Run Tests tab, as shown in Figure 1-6, then click the big green Execute Check-Mate button, as shown in Figure 1-7 (okay, so the book's color blind).**

   For your first time, just leave the options on this tab in their default state.

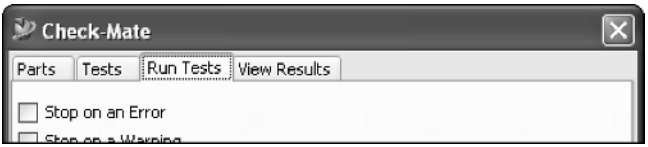

Figure 1-6: The third tab is used to Run Tests (hence the name.)

Figure 1-7: The Execute Check-Mate button is near the bottom of the Run Tests tab.

7. **Examine the results in the View Results tab.**

   Once the checks have completed running, the Check-Mate dialog automatically advances to the **View Results** tab. The tree here describes the errors found in this part. (See Figure 1-8.)

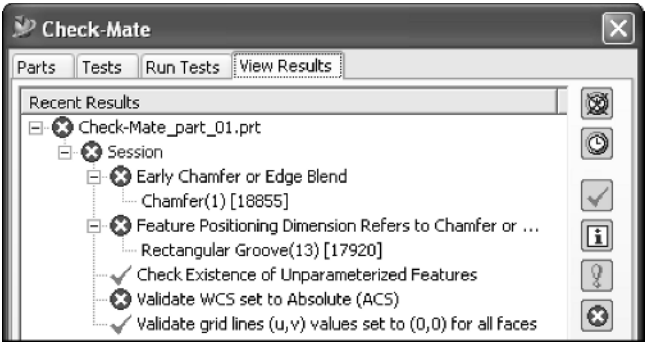

Figure 1-8: Expand the results tree to view any errors.

You can jump quickly between the errors in a long list of checks by using the VCR-style buttons in the center of the dialog (see Figure 1-9).

Figure 1-9: Use the VCR-style buttons to quickly navigate between errors.

The **Early Chamfer or Edge Blend** error is the first error listed (refer to Figure 1-8). You may be wondering, "Just what is this check looking for, anyway?"

If you want to find out what a check really does, just right-click on the check in the tree and choose **Show Test Documentation** from the pop-up menu. A window appears containing the documentation for the check or profile.

The **Early Chamfer or Edge Blend** check is looking for any chamfer or blend features appearing "early" in the model history. As a modeling best practice, many companies want their users to model all blends and chamfers toward the *end* of the model history, because if changes need to be made later, it makes the model somewhat easier to edit.

Notice that in Figure 1-8 **"Chamfer (1)"** is listed in the results tree under this check, identifying it as the offending chamfer that is "too early" in the model. The number in parentheses indicates the position in the part history, as shown in Figure 1-10.

Wait, wait, wait . . . You've just found your first problem using Check-Mate. Don't you just feel all warm inside?

8. **To fix this issue, cancel out of the Check-Mate dialog, and open the Part Navigator (see Figure 1-10). Confirm that you are in the Modeling application (Application➪Modeling) and just drag the early chamfer feature to the bottom of the model history.**

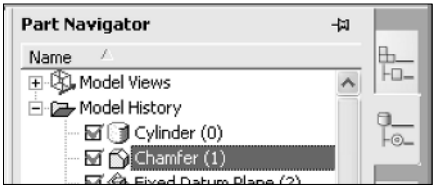

Figure 1-10: The offending chamfer feature shown in the Part Navigator.

To confirm that errors have been fixed, you can just run Check-Mate again.

9. **Click the Run Tests icon again on the Check-Mate toolbar (or from the pull-down menus, choose Analysis➪Check-Mate➪Run Tests).**

   Note that your selected checks are still visible in the **Tests** tab, and that the original results have been saved in the **View Results** tab.

10. **To re-run the tests, just choose the Run Tests tab and click the Execute Check-Mate button again.**

    If all goes well, your error is corrected, and you'll see a green checkmark displayed next to the corrected check, as seen in Figure 1-11.

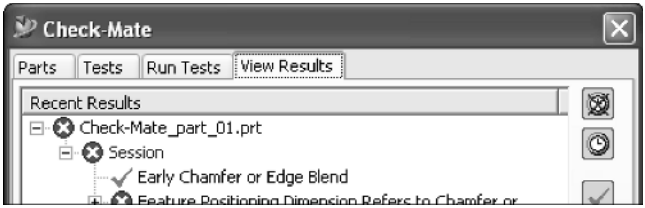

Figure 1-11: That's better. The Early Chamfer or Edge Blend check passes!

So, that's it. Choose parts, choose checks, run the checks, and then see if anything needs to be fixed. Piece of cake, right? The only thing that might get a little tedious is always making sure that you've grabbed all the right checks. Hmmm. . . .

In this next section, you see how to run the exact same set of checks against the same part, but do it using a Check-Mate *profile* instead.

What Is a "Profile"?

Most companies have already established some design standards, and ask their users to consistently validate that their designs meet the pre-defined set of company-specific criteria. UGS has provided a variety of tools for making this process easier. One such tool is the concept of a Check-Mate *profile*.

In Check-Mate, a *profile* is a collection of checks that you want to execute together at the same time. Checks contained in a profile can be pre-configured with any necessary default values or needed input parameters. A profile is a great tool for ensuring that a complete set of checks is performed using the correct set of quality criteria. Also, because it pre-configures all of the required checks in one place, a profile also simplifies the user experience a lot — and simple is good, right?

1. **In the Tests tab of the Check-Mate dialog, browse the tree and find the profile you want to use.**

In the tree on the Tests tab, individual checks are shown using black text. Profiles are shown using blue text. Category labels appear in gray. You can't really see the colors in this black and white book, but Figure 1-12 shows the configuration (and you can see the colors on your computer screen).

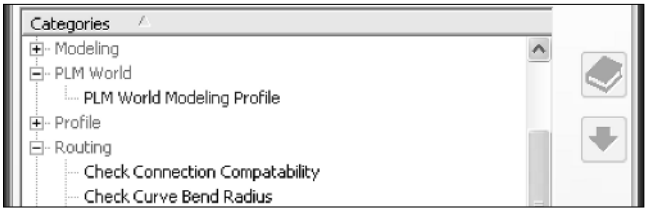

Figure 1-12: Color-coded labels in the tree on the Tests tab.

2. **Add the profile to the list of Chosen Tests by clicking the Add to Selected button (refer to Figure 1-4).**
3. **To run the selected profile against the selected part(s), you use the same Execute Check-Mate button on the Run Tests tab, of course (refer to Figure 1-7).**
4. **Examine the results.**

   Once the profile has finished running, Check-Mate automatically displays results for both the profile (as a whole) and for each individual check contained in the profile, as shown in Figure 1-13.

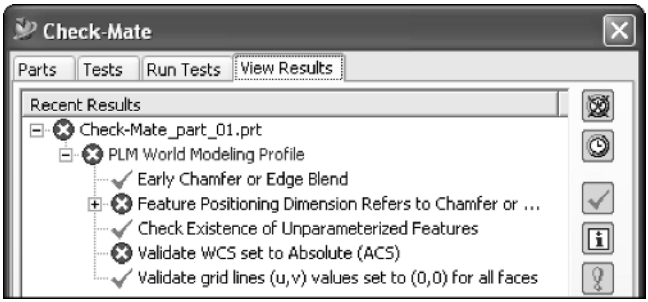

Figure 1-13: The expanded error tree after running a profile.

After selecting a check or profile, you can always press the Documentation button to the right of the tree (shown in Figure 1-14) to find out more about what that check does or which checks are included in a profile. Handy!

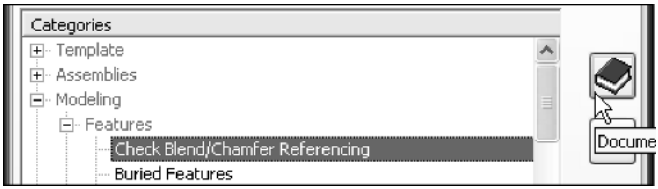

Figure 1-14: That handy Documentation button.

Using a profile simplifies the interaction needed to find, configure, and execute a set of desired checks because all the checks with their correct settings are consolidated into one item in the Tests tab. If you want everyone to be checking their models against a consistent standard, profiles really make it easy.

Faster! Faster!!

Okay, so you can see how having a pre-configured profile handy would be a lot faster than collecting individual checks, right? You wanna go even faster, you say?

The fun part about Check-Mate is that profiles are just the first step down an incremental path leading toward fully automatic validation of your designs. Part II takes you past the major milestones along that path, and you find out how to take advantage of each new step along the way.

Is taking a little bit of time to configure Check-Mate further worthwhile? I certainly think so. As you configure your Check-Mate environment, you'll be simultaneously speeding up the process of validating your models while removing opportunities for error — and that's what continuous improvement is all about, right?