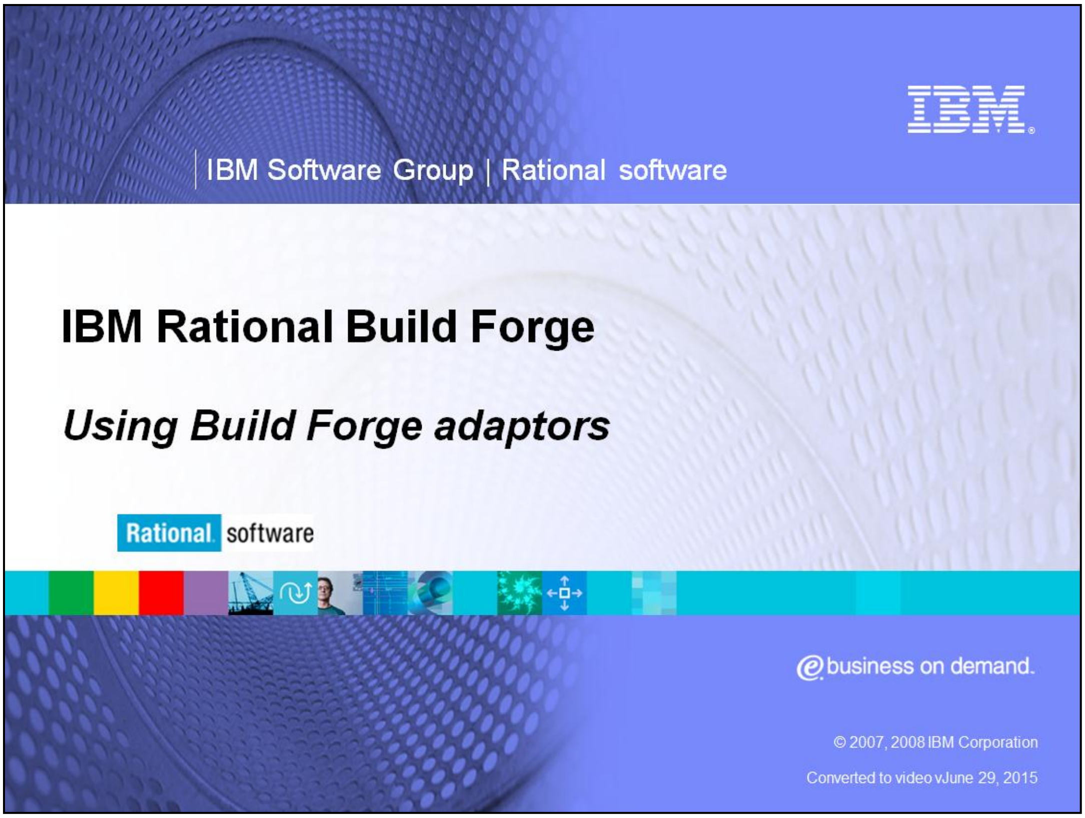


IBM Software Group | Rational software

IBM Rational Build Forge

Using Build Forge adaptors

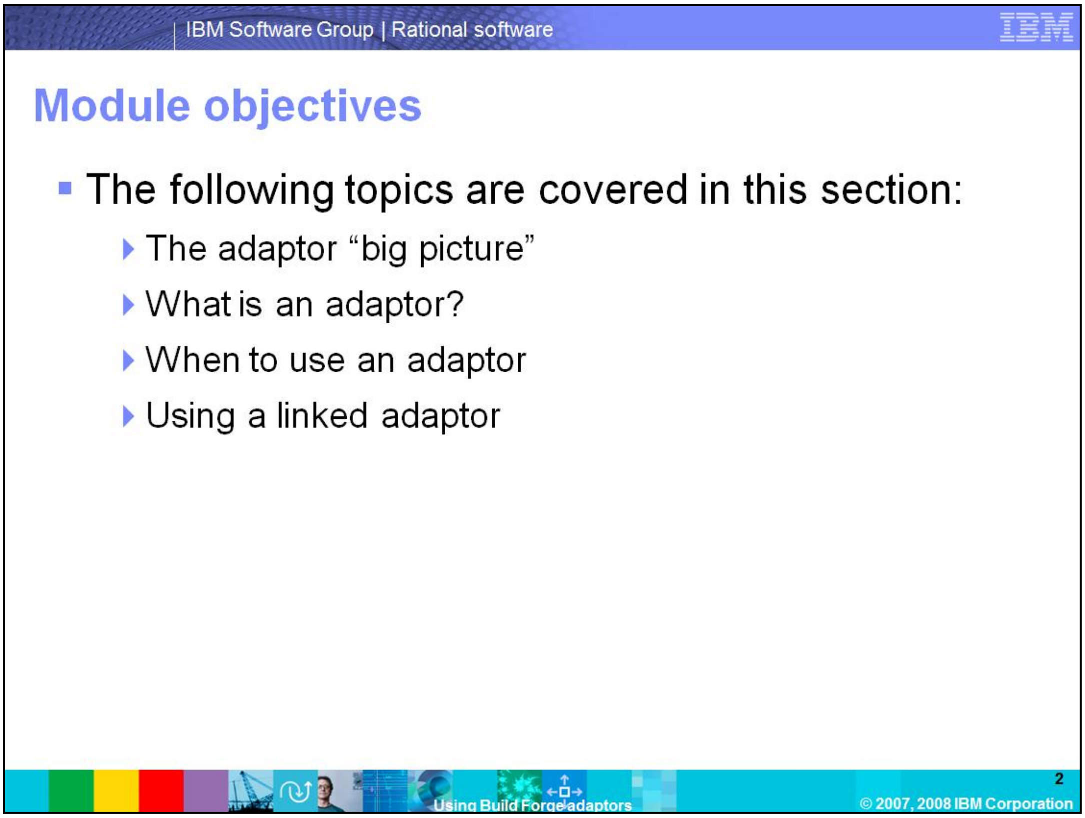
Rational software

@business on demand.

Converted to video vJune 29, 2015

IBM Rational® Build Forge® provides complete build management. IBM Rational Build Forge enables high-performance builds and agile software development. Its adaptive build-and-release management framework helps development teams standardize repetitive tasks, manage compliance mandates, and share information.

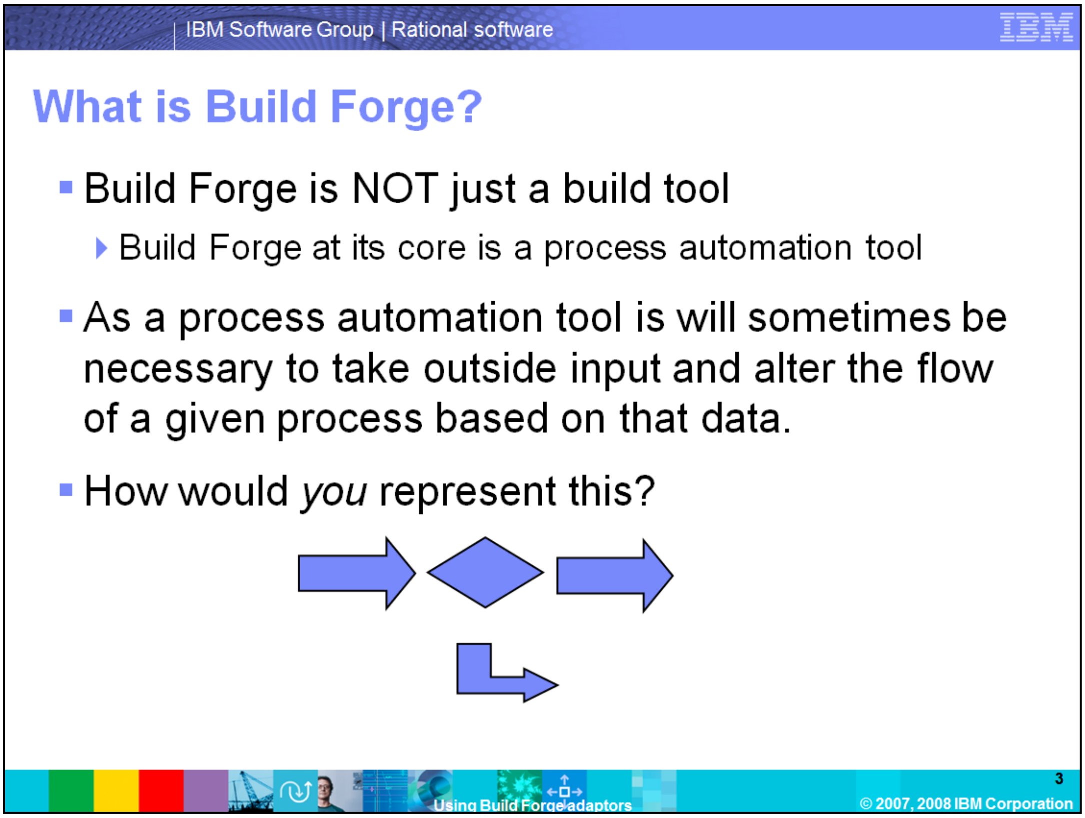
This presentation will cover an introduction to using adaptors in Build Forge.



Module objectives

- The following topics are covered in this section:
  - The adaptor “big picture”
  - What is an adaptor?
  - When to use an adaptor
  - Using a linked adaptor

This presentation will cover Build Forge adaptor functionality. There will be explanations about where adaptors fit into the overall picture of Build Forge, including what adaptors are, when to use them, and when to use linked adaptors. By the end of this presentation you should have a better understanding of what adaptors are and when it would be appropriate to use them.



IBM Software Group | Rational software

What is Build Forge?

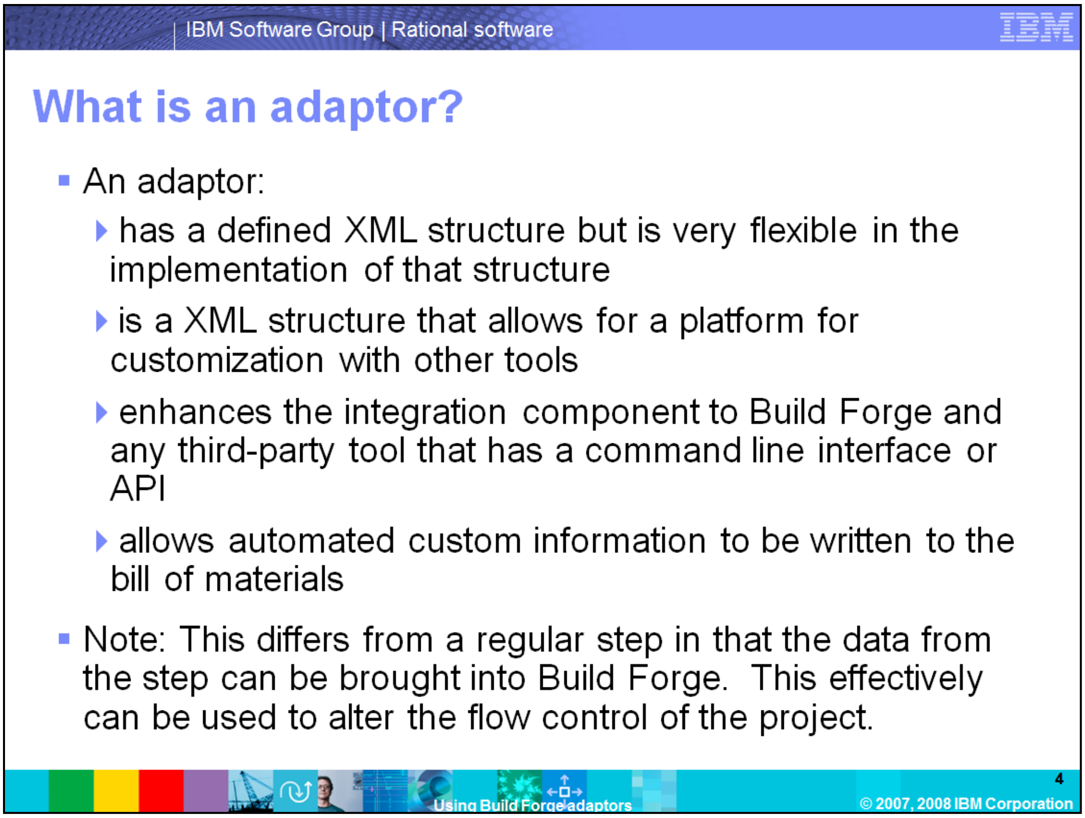
- Build Forge is NOT just a build tool
  - Build Forge at its core is a process automation tool
- As a process automation tool is will sometimes be necessary to take outside input and alter the flow of a given process based on that data.
- How would *you* represent this?

Using Build Forge adaptors

3

© 2007, 2008 IBM Corporation

Build Forge is more than a just a build tool; it is a process automation tool. This means that Build Forge does more than just run through a linear set of steps to either completion or failure; Build Forge must be able to handle branching. Specifically, this means that Build Forge will have to make a decision as to what action to take given the information available. In order for Build Forge to properly automate a process, it must be able to handle logical branches and loops. Adaptors, as discussed in this module, are the way to address this need.



IBM Software Group | Rational software

What is an adaptor?

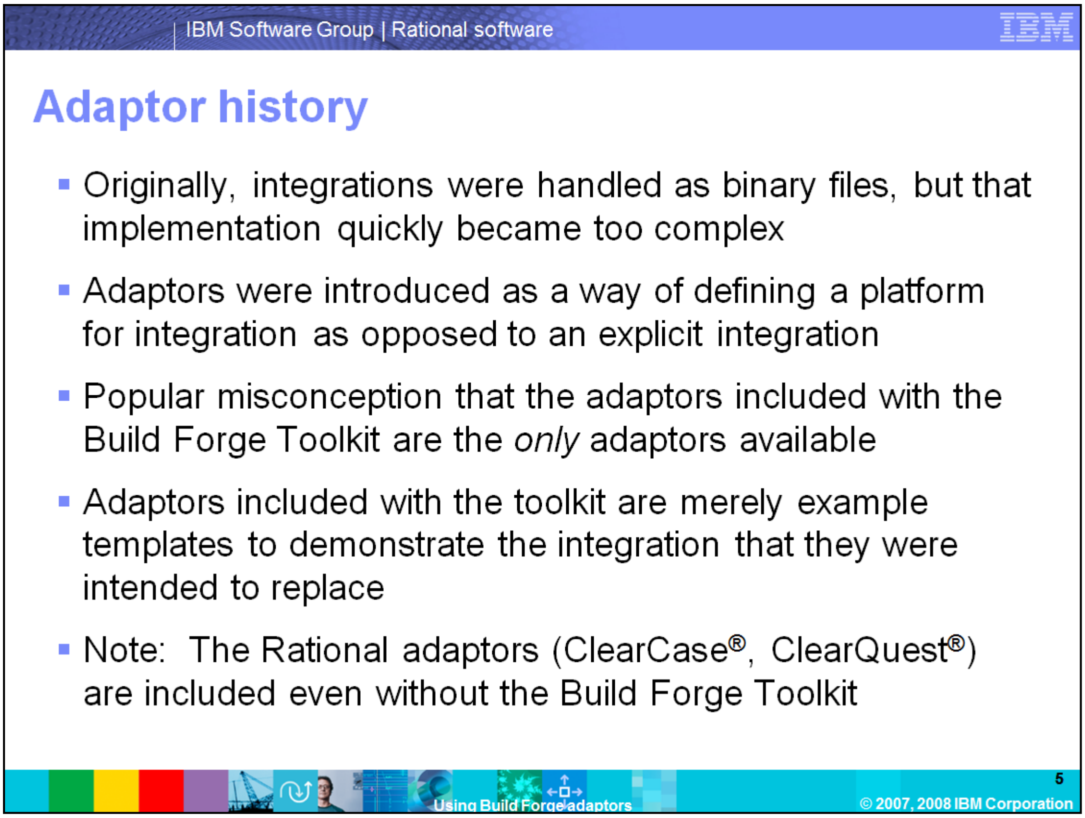
- An adaptor:
  - has a defined XML structure but is very flexible in the implementation of that structure
  - is a XML structure that allows for a platform for customization with other tools
  - enhances the integration component to Build Forge and any third-party tool that has a command line interface or API
  - allows automated custom information to be written to the bill of materials
- Note: This differs from a regular step in that the data from the step can be brought into Build Forge. This effectively can be used to alter the flow control of the project.

Using Build Forge adaptors

4

© 2007, 2008 IBM Corporation

Build Forge has a defined XML structure and syntax that must be followed, but it is very flexible in the implementation of that structure. It is an XML structure that allows for a platform for customization with other tools; it enhances the integration component to Build Forge and any third-party tool that has a command line interface or API; and it allows automated custom information to be written to the bill of materials. The adaptor can effectively be thought of as a Build Forge specific scripting language. Note that this differs from a regular step in that the data from the step can be brought into Build Forge. This effectively can be used to alter the flow control of the project.



IBM Software Group | Rational software

Adaptor history

- Originally, integrations were handled as binary files, but that implementation quickly became too complex
- Adaptors were introduced as a way of defining a platform for integration as opposed to an explicit integration
- Popular misconception that the adaptors included with the Build Forge Toolkit are the *only* adaptors available
- Adaptors included with the toolkit are merely example templates to demonstrate the integration that they were intended to replace
- Note: The Rational adaptors (ClearCase®, ClearQuest®) are included even without the Build Forge Toolkit

Using Build Forge adaptors

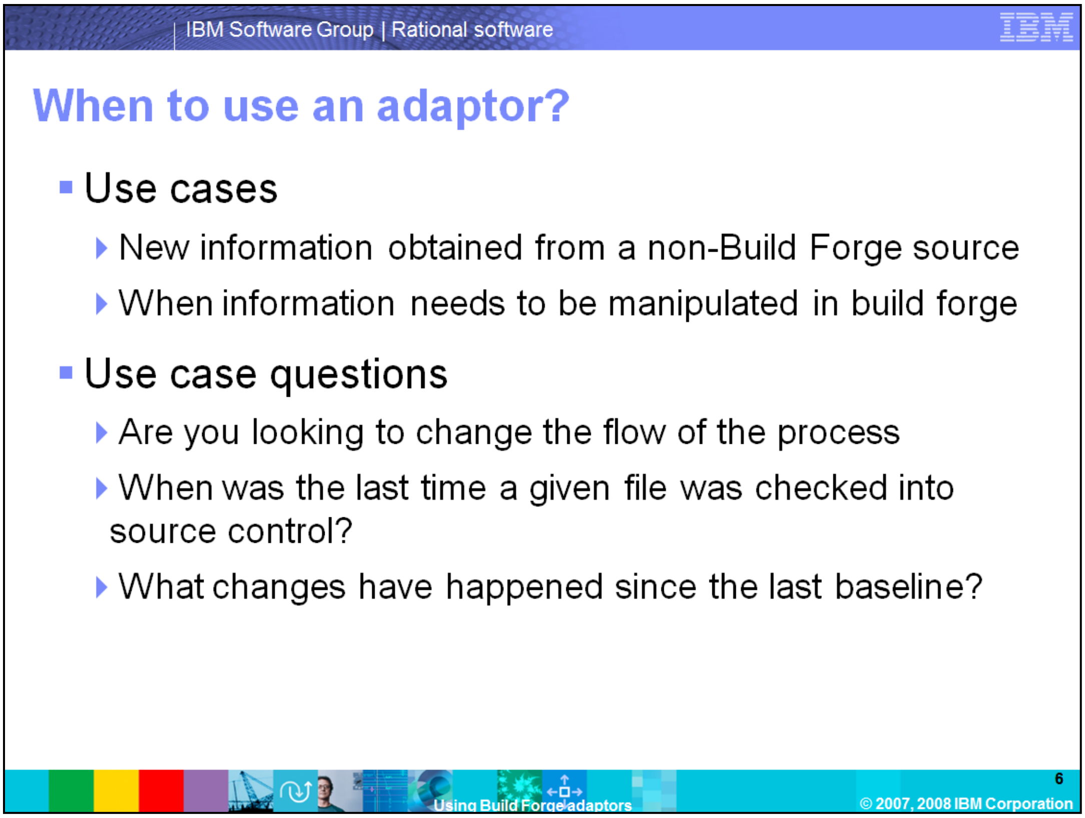
5

© 2007, 2008 IBM Corporation

Build Forge adaptors currently are created to allow users flexibility in use. In earlier versions of Build Forge the adaptors were in binary form and Build Forge was responsible for making these complex integrations work correctly.

Adaptors were introduced as a way of defining a platform for integration as opposed to an explicit integration. With the current implementation of an XML framework, Build Forge adaptors allow greater flexibility and the additional functionality to define logical flow. It is important to note that there is no longer a single-tasked idea as to what adaptors should do; just a flexible framework.

It is a popular misconception that the Adaptors included with the Build Forge Toolkit are the only Adaptors available. The truth of the matter is those adaptors are example templates created to demonstrate the integration that they were intended to replace. Note that the Rational adaptors for ClearCase and ClearQuest are included even without the Build Forge Toolkit.



IBM Software Group | Rational software

When to use an adaptor?

- Use cases
  - New information obtained from a non-Build Forge source
  - When information needs to be manipulated in build forge
- Use case questions
  - Are you looking to change the flow of the process
  - When was the last time a given file was checked into source control?
  - What changes have happened since the last baseline?

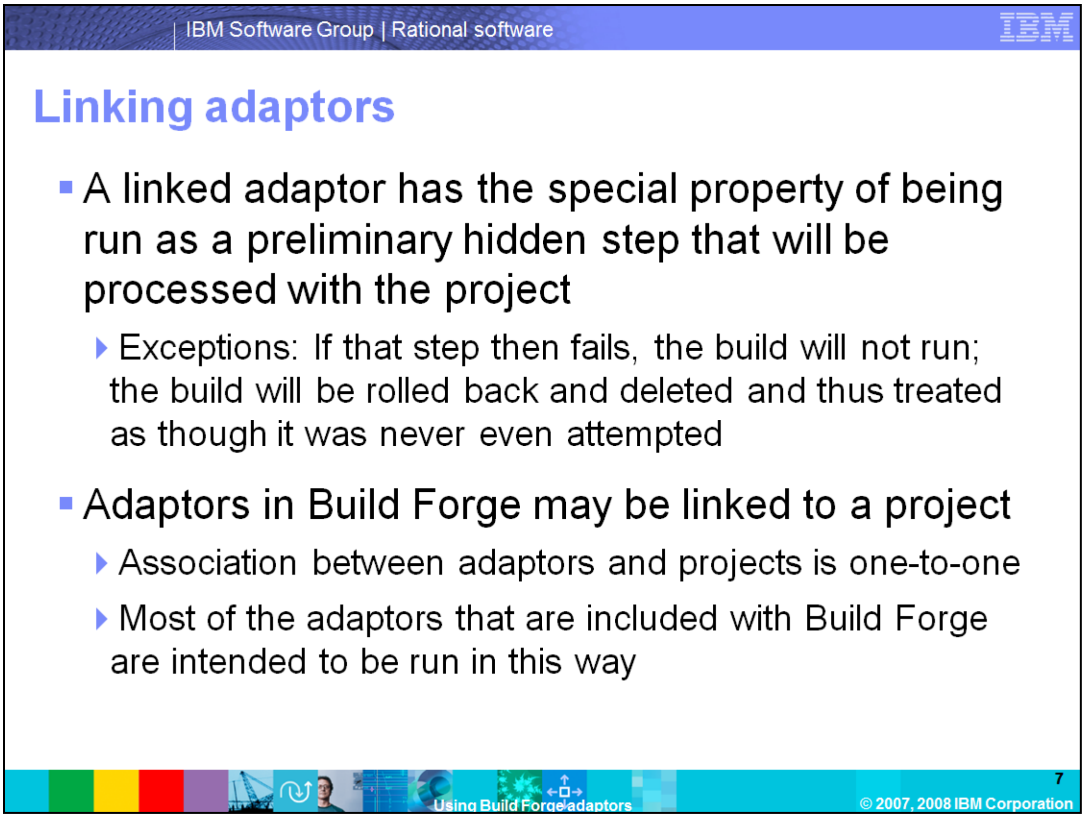
Using Build Forge adaptors

6

© 2007, 2008 IBM Corporation

Adaptors should be used whenever outside information must be brought in or manipulated in Build Forge. When you are considering use cases, ask yourself if you are looking to change the flow of the process based on information stored outside of Build Forge Adaptors are the *method* to take that information into Build Forge, and Build Forge is used to make a logical flow decision based on that information. If you find yourself explaining the use case with an IF or ELSE, then Adaptors will be the functionality you need to use.

Generally speaking, whenever the use case involves getting new information from a non-Build Forge source, the answer will be accomplished through an adaptor.



IBM Software Group | Rational software

Linking adaptors

- A linked adaptor has the special property of being run as a preliminary hidden step that will be processed with the project
  - Exceptions: If that step then fails, the build will not run; the build will be rolled back and deleted and thus treated as though it was never even attempted
- Adaptors in Build Forge may be linked to a project
  - Association between adaptors and projects is one-to-one
  - Most of the adaptors that are included with Build Forge are intended to be run in this way

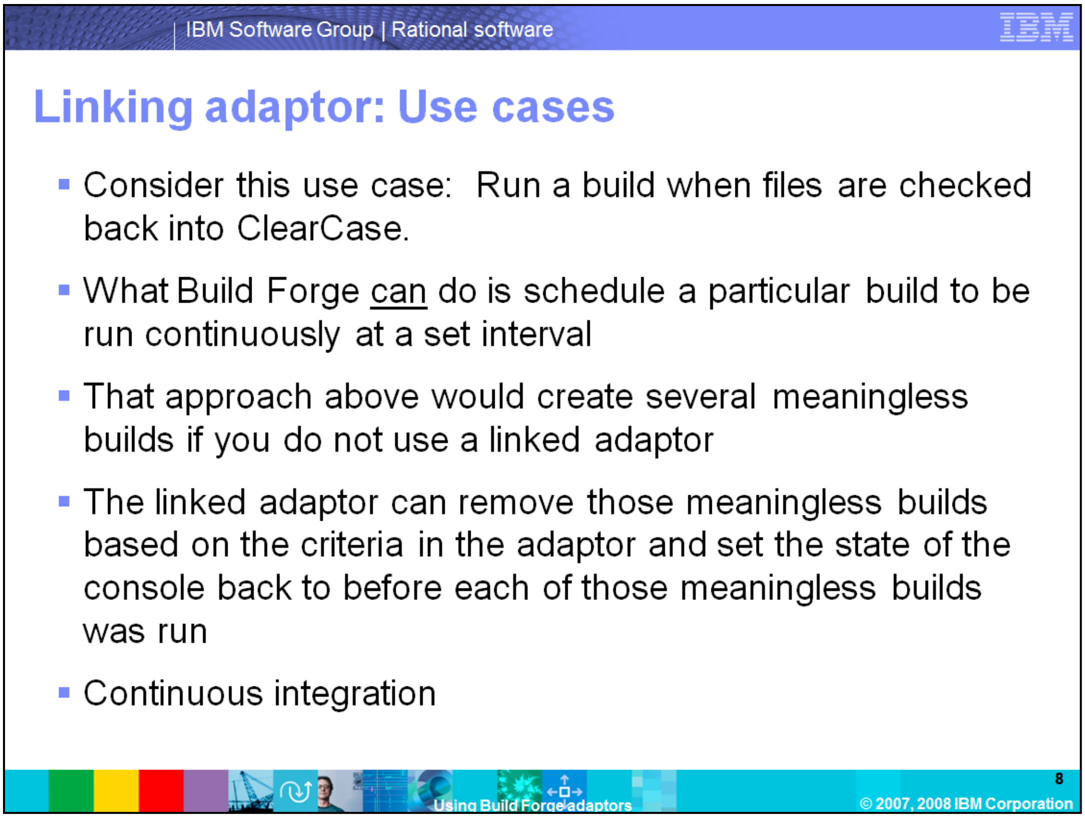
Using Build Forge adaptors

7

© 2007, 2008 IBM Corporation

Linked Adaptors are a special use case that apply to those situations where you are trying to decide whether or not to run a build. A linked adaptor runs as a step before the build actually runs. If this preliminary step fails then the build is completely removed from the database. The build will be rolled back and deleted, and essentially be treated as though it was never even attempted. That is to say there is no record that the build ever took place, so the next execution will run as though there was no build before it.

Adaptors in Build Forge may be linked to a project. Currently, association between Adaptors and projects is one-to-one, and note that most of the Adaptors that are included with Build Forge are intended to be run in this way.



IBM Software Group | Rational software

Linking adaptor: Use cases

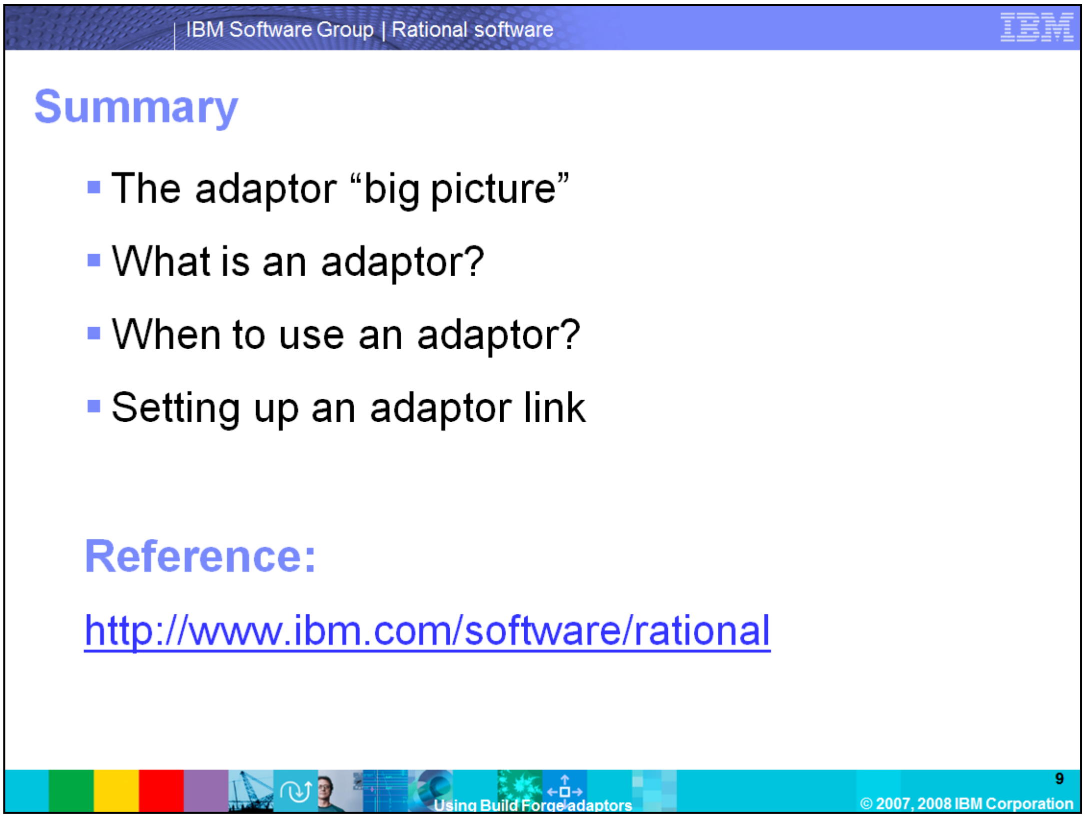
- Consider this use case: Run a build when files are checked back into ClearCase.
- What Build Forge can do is schedule a particular build to be run continuously at a set interval
- That approach above would create several meaningless builds if you do not use a linked adaptor
- The linked adaptor can remove those meaningless builds based on the criteria in the adaptor and set the state of the console back to before each of those meaningless builds was run
- Continuous integration

Using Build Forge adaptors

8

So, what is the value of linking the adaptor? Consider the use case of running a build when files are checked back into ClearCase. Strictly speaking, Build Forge itself cannot do that, it needs adaptors to help.

The intention of adaptors was to implement “continuous integration.” So if you have a build with a linked adaptor that is set to run every 5 minutes, then only those builds that pass the adaptor will actually get built. Without the linked adaptor there would be a bunch of meaningless builds that were each essentially the same as the build before it. This is “continuous integration” because the moment something happens on the system that should generate a new build, there should be a scheduled build that will run that build and adaptor.



IBM Software Group | Rational software

Summary

- The adaptor “big picture”
- What is an adaptor?
- When to use an adaptor?
- Setting up an adaptor link

Reference:

http://www.ibm.com/software/rational

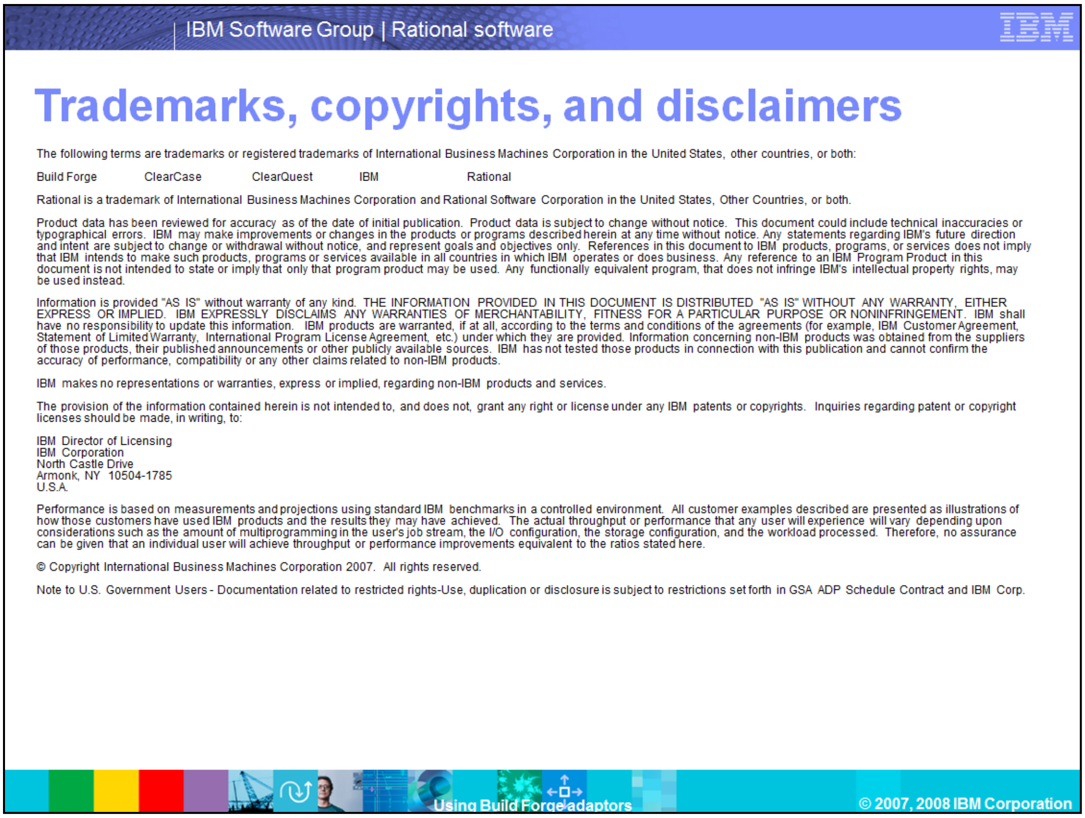
Using Build Forge adaptors

9

© 2007, 2008 IBM Corporation

In summary, you should now have a better understanding of what adaptors are and when to use them in Build Forge. You should be able to ask yourself key use case questions as discussed in this module to ensure you are using adaptors or linking adaptors at the right time. The next module provides examples of adaptors and syntax.

For more information on using adaptors in IBM Rational Build Forge, visit http://www.ibm.com/software/rational .



Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

Build Forge ClearCase ClearQuest IBM Rational

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.