
Inform 7 Feature User's Guide

Table of Contents

Preface	ii
Installation	1
Prerequisites	1
Java	1
Eclipse	1
Inform 7	1
Installing the Inform 7 Feature	1
Uninstalling the Inform 7 Feature	2
Getting Started	4
Additional Documentation	4
Customizing the Resource Perspective	4
Creating an Inform 7 Project	5
Editing and Compiling Inform 7 Code	6
Running a Compiled Story	7
Skein Editor and Transcript View	8
Comparison with the Inform 7 IDE	11
Known Bugs & Problems	13
Acknowledgements	14

Preface

The Inform 7 Feature for Eclipse is a set of plug-ins which will support cross-platform development of interactive fiction projects in the Inform 7 language. However, the current version is only tested under Linux.

Inform 7 for Windows and OS X ships with a standalone graphical Integrated Development Environment (IDE), which is described in the official documentation. However, this IDE is not available for Linux (yet); only a simple text-based interface exists so far. As an alternative, the Inform 7 feature for Eclipse was created and released as an open source project.

Thanks to the portability of the Eclipse and Java™ technology, the software may (in the future) also work under Windows, OS X and Solaris, with reduced need for developers to maintain a separate entire project for each platform. Eclipse's built-in features such as version control and tight integration with other tools make the Inform 7 Feature potentially more powerful than the standalone Inform 7 IDE. On the other hand, some of the nice features of the Inform 7 IDE remain unimplemented, still.

The overall approach taken during the development of the Inform 7 Feature is to treat Inform as yet another programming language, among the many programming languages already supported by the Eclipse platform. This approach seeks to reuse many concepts and user interface metaphors developed over the years in the domain of integrated development environments. It differs somewhat from the approach followed by Inform 7 itself, which presents its IDE more like a specialized word processing application than a familiar programming environment.

The Inform 7 Feature for Eclipse project follows the so-called "bazaar" (rapid prototyping) style of development - early, frequent releases under a liberal license (Eclipse Public License v. 1.0). Public forums are available for discussing bugs and improvements and external contributions in form of patches and ideas are welcome. Time will tell whether the Inform 7 Feature for Eclipse will mature to become a serious alternative to the default Inform 7 IDE or become obsolete by a Linux port of that IDE.

Installation

Prerequisites

Java

Eclipse, and consequently the Inform 7 feature, requires a Java Runtime Environment (JRE). This component has to be installed on your machine. You will need at least version 1.4.2. To find out whether Java is already installed, you can run the command `java -version` (in a Unix shell or in the DOS box, depending on your platform). If necessary, you can obtain the current version of Java directly from www.java.com [<http://www.java.com>].

Eclipse

The minimum requirement for the Inform 7 feature is version 3.2 of Eclipse. If you already have Eclipse installed, you can skip this step.

Eclipse can be downloaded from www.eclipse.org [<http://www.eclipse.org>]. It comes in different flavors (with different sets of preinstalled features). For using the Inform 7 feature, any distribution which includes the Update Manager is sufficient. Installing Eclipse is easy. Just download it and unzip it somewhere. The executable file `eclipse` (or `eclipse.exe`) is ready to run after that.

The first time when Eclipse is started, it will ask you for the location of the "workspace" directory. Workspace is where all your projects as well as some additional data are usually kept. Choose any directory on your hard disk to which you have write access.

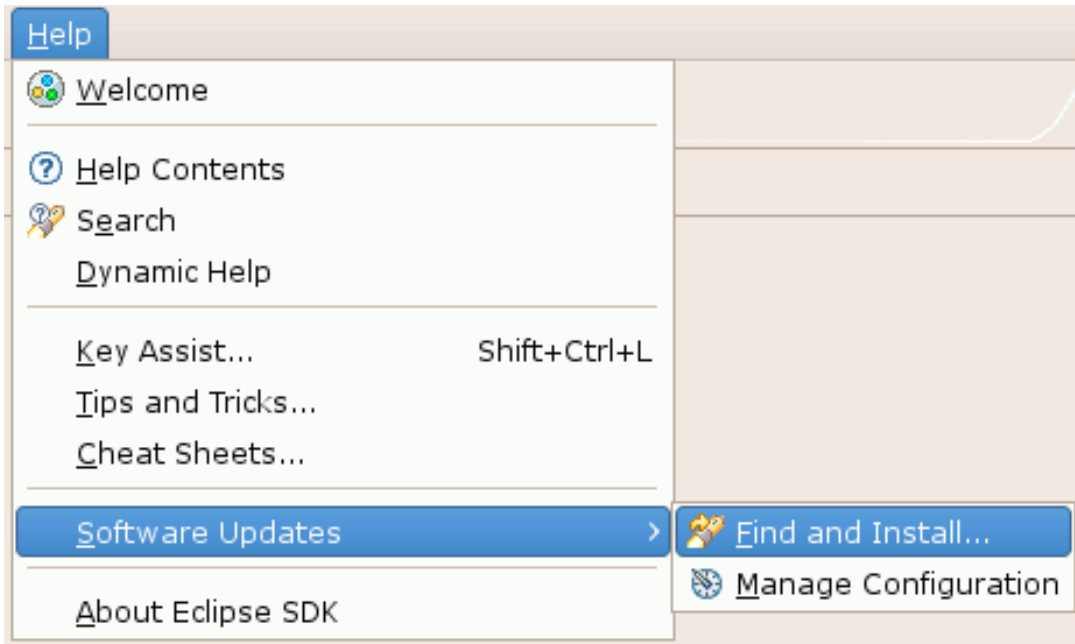
Inform 7

The Inform 7 feature for Eclipse requires the compilers which ship in the Inform 7 distribution; they have to be installed separately. Download and install the appropriate version for your platform from www.inform-fiction.org [<http://www.inform-fiction.org>]. Take a note of the target installation directory; you will be asked for it in the final installation step.

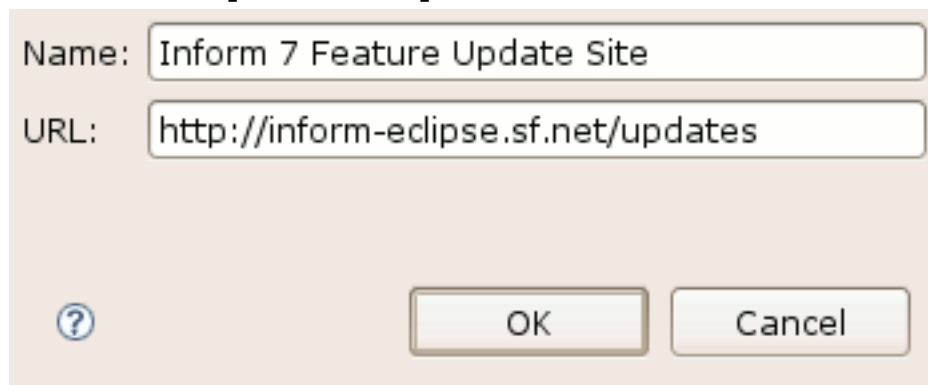
Installing the Inform 7 Feature

The installation is done from Eclipse using the Update Manager. The Update Manager connects to the project's update site [<http://inform-eclipse.sourceforge.net/updates>] and downloads the necessary files from there. The update site can also be stored locally for offline installation. It can be downloaded from the Inform 7 Feature for Eclipse project page [<http://inform-eclipse.sourceforge.net>].

After starting Eclipse, select Help # Software and Updates # Find and Install... from the menu.



Select Search for new features to install and press Next. Press the Add Update Site... button for a remote installation from the Internet or the Add Local Site... button if the Update Site is available on your local disk. When installing the feature from the Internet, enter any desired Name and the URL <http://inform-eclipse.sf.net/updates>.



The installation steps differ slightly depending on whether you have already installed the Graphical Editing Framework (GEF) feature, which is a necessary prerequisite for the Inform 7 feature. The following description assumes that you don't have GEF. If you do have it (for example because you are upgrading the Inform 7 Feature rather than installing it from scratch), you can skip the GEF-related steps:

Select the newly created Inform 7 Feature Update Site and Eclipse's predefined Europa Discovery Site and press the Next button. On the next page, select the Inform 7 Feature Update Site, and then expand the Europa Discovery Site and press the Select Required button. This will automatically select the Graphical Editing Framework (GEF) feature from the Europa Discovery Site. Follow the instructions of the installation wizard. The warning about the installation of an unsigned feature can be ignored. Eclipse has to be restarted after the installation.

After Eclipse is restarted, open the Window # Preferences... dialog and select Inform 7 in the left pane. Enter the path to the directory into which you previously installed the Inform 7 distribution (see the section called "Inform 7") and close the dialog by clicking OK. The installation is now complete.

Uninstalling the Inform 7 Feature

To temporarily disable or uninstall the currently active version of the feature from Eclipse, select

Help # Software and Updates # Manage Configuration from the menu. Expand the tree in the dialog window which appears and select the Inform 7 feature. Click on the option Disable in the panel on the right side and restart Eclipse when asked. At this point the feature is disabled, but still present on disk. To remove it completely, enter the same dialog again and select the previously disabled version of the feature (you may need to toggle showing disabled features in the dialog's toolbar). The option Uninstall will now remove the chosen version of the feature completely.

Getting Started

This chapter will walk you through creating, compiling and launching your first Inform 7 project. However, let's review some additional sources of information first.

Additional Documentation

The remaining online documentation beyond this chapter contains a complete reference of the functionality of the Inform 7 Feature for Eclipse.

Note that the online documentation you are reading now does not explain the Inform 7 syntax or programming concepts, just the IDE functionality. If you are new to Inform 7 programming, you should also spend some time reading the official language documentation [<http://www.inform-fiction.org>].

If you are new to Eclipse, you will also benefit from reading the Workbench User Guide (part of Eclipse's online help), which describes the main concepts shared by most Eclipse plug-ins. A good grasp of these concepts speeds up learning new Eclipse features considerably. In fact, if you are already familiar with Eclipse, you probably do not need to study much of the remaining documentation because the Inform 7 feature follows Eclipse's user interface conventions.

Customizing the Resource Perspective

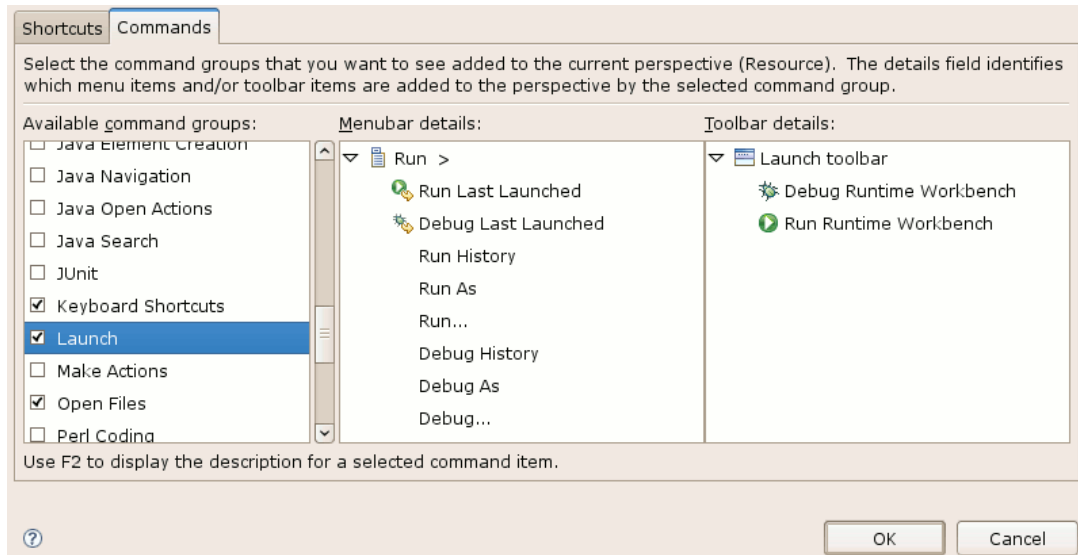
Eclipse presents its graphical interface using so-called perspectives. The currently active perspective (there can be only one for a window) determines which elements of the user interface are visible. The overall idea is that the readily available functionality should reflect your current task (e.g. writing Inform 7 code, programming in Java, or browsing a database). The Inform 7 Feature does not define an own perspective yet, so we will use the (slightly customized) default Resource perspective in this tutorial. Proceed as follows:

1. Select Window # Open Perspective # Other... from the main menu.
2. Select Resource in the dialog that appears and click OK.

Depending on your previous perspective, the layout of your screen might become adjusted somewhat. For example, the Navigator view should be visible now. The next step is to add some other views and functions which we will use in this tutorial to the perspective. Right now, we are interested in the following views: Navigator, Outline, Console, and Problems. To open these views (unless they are already open), select Window # Show View # Other... from the main menu and pick the appropriate view from the dialog. Note that you can close all other views to reduce clutter.

The last remaining step is to add some necessary commands to the menu Run in the menu bar. Proceed as follows:

1. Select Window # Customize Perspective... from the main menu.
2. In the dialog which appears switch to the Commands tab.
3. Put a tick next to Launch in the Available command groups list.
4. Click OK to close the dialog.



After these steps, the menu Run in the main menu bar should now contain some interesting items which we will examine later.

Creating an Inform 7 Project

Each interactive fiction story is managed within a separate project in Eclipse. A project is a directory tree typically rooted in the workspace directory of Eclipse. You can create a new Inform 7 Project as follows:

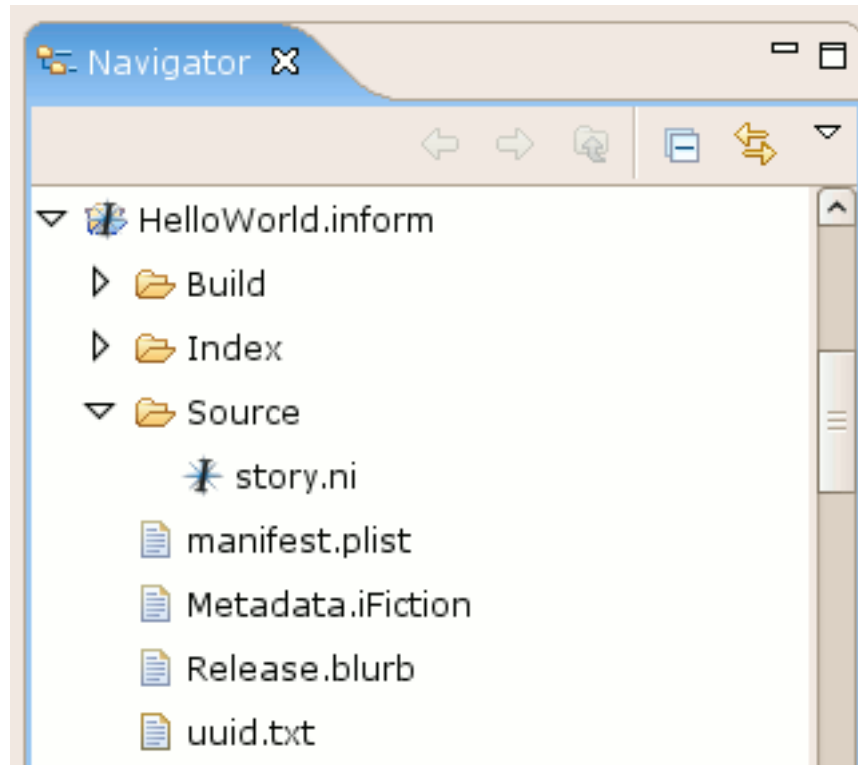
1. Select File # New # Project... from the main menu.
2. Select Inform 7 Project from the list and click Next.
3. Enter a project name. By convention, Inform 7 projects carry the suffix `.inform` in their name. For example, enter **HelloWorld.inform** as the project name. The Location field by default points to a corresponding project directory within the workspace. If you wanted your project directory to reside elsewhere, you could enter an alternative location manually. For example, this would be necessary if you wanted to import an existing Inform 7 project which you created earlier outside of the workspace.
4. Click on Finish to create the project (or import an existing one from the specified location).

Inform 7 Project

Create a new Inform 7 project.

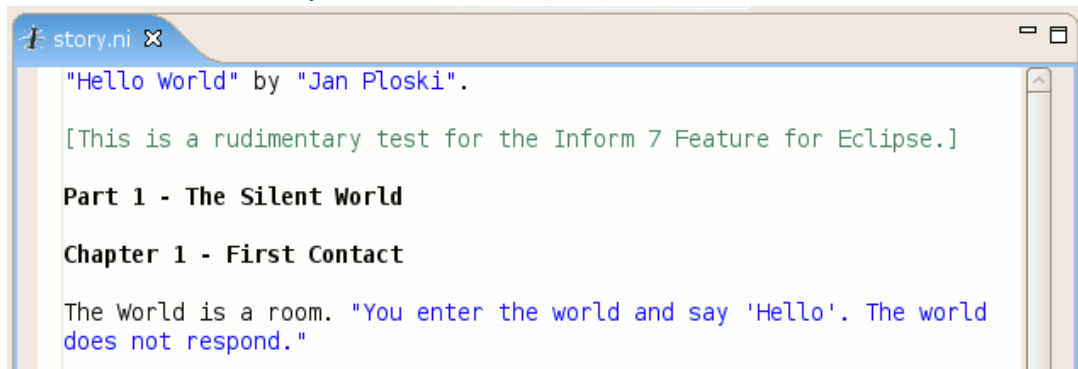


The Navigator view should now display the newly created project. Click on the little arrow left of the project name to show its contents:



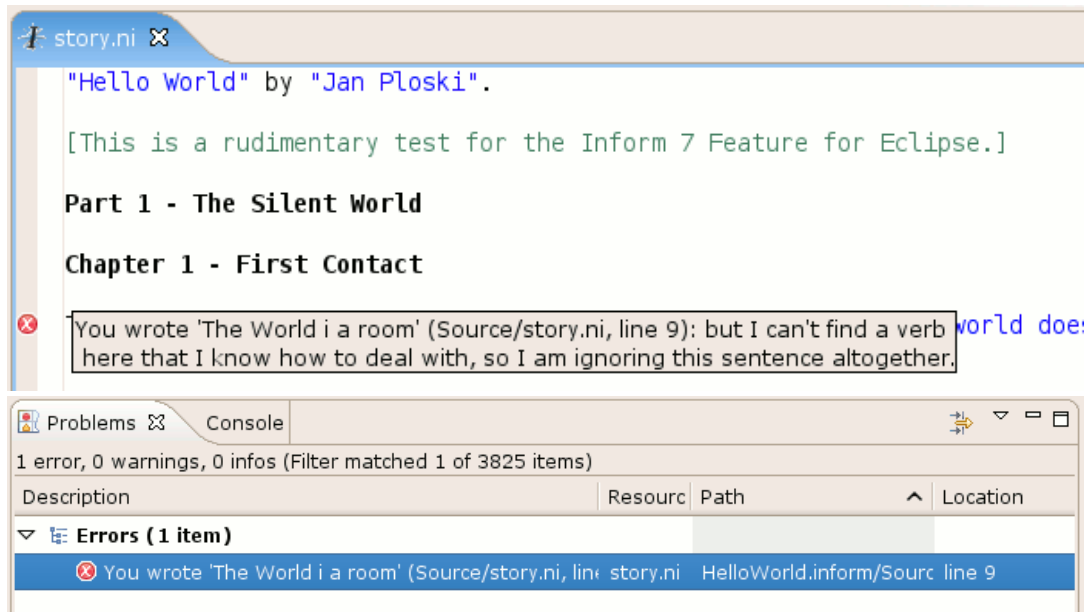
Editing and Compiling Inform 7 Code

Double-click on the file `Source/story.ni` in the Navigator view to open it in the Inform 7 editor. You will notice that the file contains a placeholder for the story title and author. Enter some more text for the HelloWorld story:



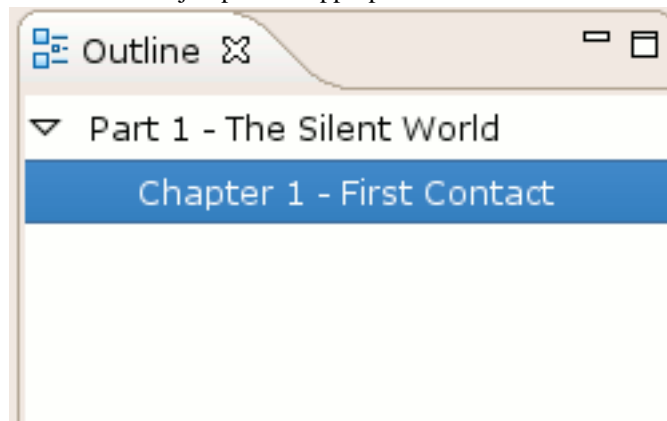
When you save the story with **Ctrl+S**, the story will likely become automatically compiled. Unless there are errors, you might not even notice. The auto-compilation behavior is governed by the setting **Project # Build Automatically** in the menu, which you can disable if you prefer to compile your story manually using the **Project # Build Project** command. In either case, the successful compilation will by default create the file `Build/output.z8` (as well as some other files).

If you make a mistake in the story and attempt to compile it, the errors will appear as red markers on the left margin of the editor as well as in the Problems view (which you opened earlier in this tutorial). Move the mouse pointer over the marker to see the full error message. Double-click an item in the Problems view to jump to the corresponding location in the editor.



Correct the mistake and save/compile again to make the error message disappear.

While you edit text in the editor, the Outline view keeps track of the overall structure. The caret location in the editor is reflected by highlighting the appropriate entry in the Outline. Similarly, clicking on an entry in the Outline will jump to the appropriate location in the editor:



Running a Compiled Story

Running a compiled Inform 7 story is similar to running any program from within Eclipse. First, you create a so-called launch configuration which describes which file to run and (optionally) how to run it (this step is typically done just once). Second, you tell Eclipse to execute your launch configuration (this step typically occurs many times). Currently, the Inform 7 Feature for Eclipse only supports running Z-code stories (with the file extension z5, z6, or z8).

To create a launch configuration for your story, proceed as follows:

1. Select Run # Run... from the menu.
2. In the following dialog, click on Inform 7 Code in the left pane and then click the New button.
3. Edit the name of your launch configuration. This can be anything you like and does not have to match the project or executable file name. To keep things orderly, enter **HelloWorld.inform**.
4. In the Project field, either enter **HelloWorld.inform** or click Browse... to select the project

from a list.

5. In the File to execute field, either enter **Build/output.z8** or click Search... to select the file from a list.
6. Because you wish to run the newly created launch configuration right away, click Run.

After you run the configuration, the dialog disappears and seemingly nothing happens. Switch to the Console view to convince yourself otherwise. Your story is displayed in this view and you can enter commands by typing after the prompt to interact with the game and see the interpreter's responses.

```

HelloWorld.inform [Inform 7 Code] IF Interpreter
World 0/1
.
Hello World
An Interactive Fiction by Jan Ploski
Release 1 / Serial number 070814 / Inform 7 build 4U65 (I6/v6.31 lib 6/11N)
SD
World
You enter the world and say "Hello". The world does not respond.
> sing
World 0/2
.
Your singing is abominable.
>

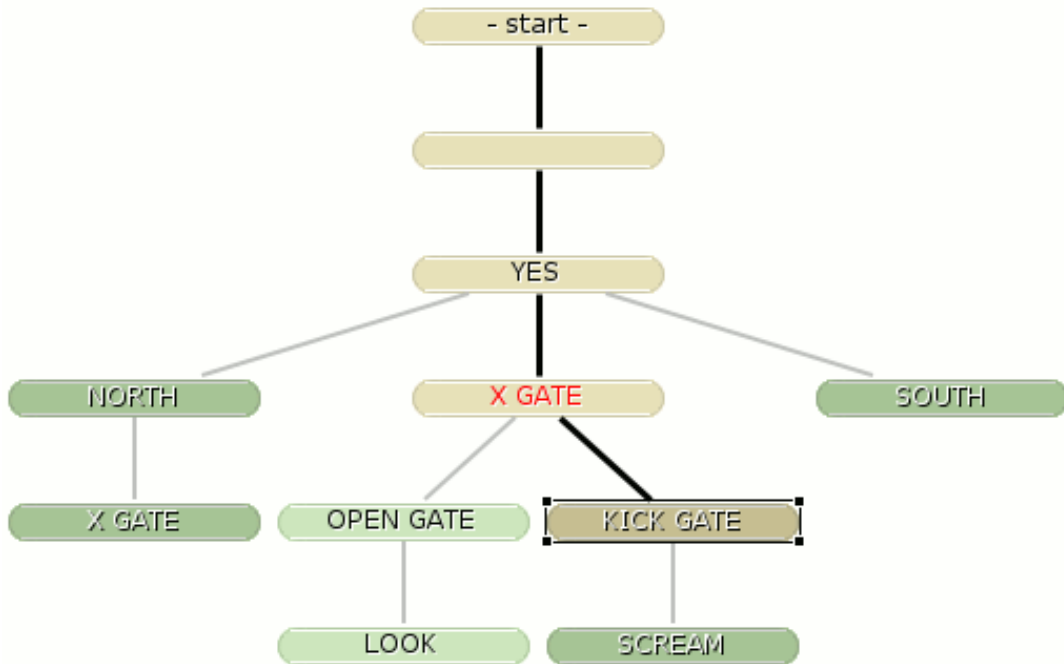
```

To terminate the interpreter, click on the red button in the Console's tool bar. Note that the Console can display multiple active launches at once (they could be multiple running Inform stories, or even other programs; you can flip between them using the pulldown menu). The terminate button always terminates only the currently displayed launch. After the launch is terminated, it can be removed from the Console view with the "X" button.

To run your story again, you could select Run # Run... from the menu. However, this is bothersome in the long term. It is more convenient to use the green Run icon from the main toolbar which remembers the most recently launched configurations as a shortcut. Just clicking this icon will re-launch the last configuration. Clicking on the little arrow next to it will open a pulldown menu with a selection of launch configurations.

Skein Editor and Transcript View

When testing a story systematically, it is useful to create transcripts that record the different possible transcripts. A transcript is simply a sequence of commands that the player might issue, along with the corresponding game responses. Because multiple transcripts may start with the same prefix (the same initial in-game commands), it is useful to visualize them as a tree or a "skein", in which the different transcripts correspond to different "threads" descending from a common beginning node. An example skein is shown below:

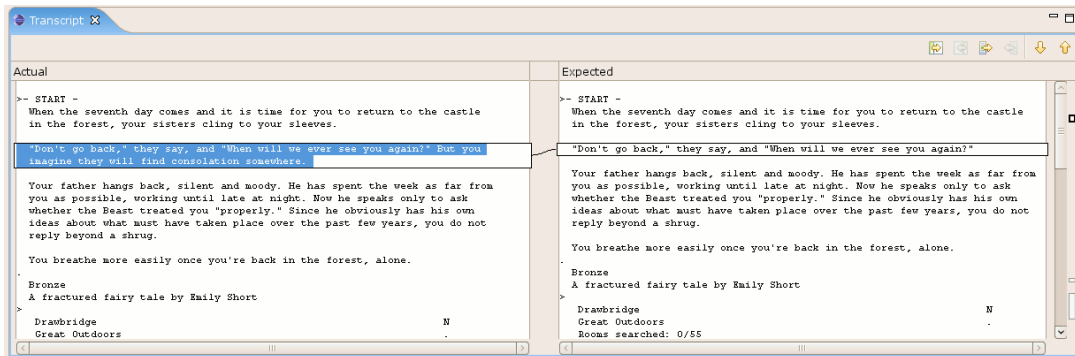


Skein nodes correspond to commands issued by the player in one or more games. A node's color informs about its current status - the golden nodes are commands issued during the current game, while the green nodes belong to the (yet) unplayed paths. Furthermore, the lighter nodes indicate that the expected game response has been already recorded by the game designer (you), while darker nodes portray commands with no expected responses. A red label indicates that the actual response from the game has been different than the recorded expected response.

To create (or update) a skein, edit an Inform 7 launch configuration (as described in the section called “Running a Compiled Story”) and enter a path to a *.skein file in the field Skein to update (the file does not have to exist yet - it will be created for you). The path is project-relative - simply **Skein.skein** will do. When you run the configuration, a skein editor will open automatically. Every command you issue in the game will be added to the skein. Moreover, the selection in the editor will follow your last issued command. When you terminate the launch configuration, the editor will close by default. However, you may re-open it by double-clicking on the newly created *.skein file. Multiple launch configurations may point to the same skein file, although only one active launch configuration is allowed to update it at a time. If you like, you may also create multiple skein files in your project.

In the skein editor, right-clicking on a node brings up the context menu with the actions Replay and Prune. Replay (also available via double-click) runs the first launch configuration associated with the skein and replays all commands up to and including the selected node. Prune removes the node, including all descendant nodes, from the skein.

To view transcripts recorded in the skein and edit the expected game responses, select Window # Show View # Other... from the main menu and in the dialog choose Transcript from the Inform 7 Skein category.



The Transcript view is linked to the active skein editor. That means that when you change selection by clicking on nodes in the editor, the contents of the Transcript are refreshed to reflect the path from the start node to the selected node. The view is divided into two panes: the left pane, titled Actual, shows the most recent game responses collected on the selected path; the right pane, titled Expected, is editable and shows the recorded expected responses. You can update the Expected pane by clicking on the Copy Current Change from Left to Right icon in the view's toolbar or by entering the text directly (this is called "blessing" of transcript contents in the original Inform 7 IDE). Take care not to destroy the structure of the transcript (commands preceding responses), or the contents will not be recorded correctly in the skein file. Your updates are saved automatically whenever you change node selection or close the skein editor.

Note

Depending on your screen resolution, it may be a good idea to change the font used in the Transcript view to a smaller size to avoid cut-off text. Go to Window # Preferences and then select General # Appearance # Colors and Fonts. Change the Compare text font to your preferred size.

Note

Captioning of paths in the skein and auto-pruning of nodes is not yet implemented.

Comparison with the Inform 7 IDE

This chapter contains a brief list of features of the official Inform 7 IDE implemented in the Inform 7 Feature for Eclipse and a list of features that are still missing.

Implemented features:

- Create Inform 7 projects
- Projects automatically managed as single folders
- Projects may be freely exchanged across supported platforms
- Each project displayed in a single window (if desired)
- Multiple, resizable views on the project open at once
- Source text editor with syntax highlighting
- Search/replace in source text
- Configurable font
- Auto-save source text on build
- Spell-checking (in Eclipse 3.3 or later)
- Entire project can be rebuilt from source text alone
- Convenient compilation (on request or automatic)
- Compilation problems automatically brought into view
- Actual output from command-line tools viewable
- Tolerate faults in command-line tools
- Lengthy scrollbar (configurable)
- Interminably looping or out-of-control games can be interrupted
- Replay last sequence of game commands
- Skein
- Auto-recording of transcripts, "blessing" of transcript contents

Unimplemented features:

- Create and manage Inform 7 extensions
- Author's (nick)name remembered across projects
- Support for legacy Inform 6 projects
- Discard intermediate build files automatically / when necessary
- Configurable font styles and color schemes
- Paragraph indentation automatically managed

- Auto-numbering of sections
- Clickback icons for jumping into source code
- Display word count, room count and item count after compilation
- Automatically (re-)start game after successful compilation, no need to type "quit"
- Command recall using the up-arrow key
- Integrated browser for the generated HTML index
- World map viewer
- Support for project publishing

Known Bugs & Problems

The Bugs [http://sourceforge.net/tracker/?group_id=75859&atid=545274] and Feature Requests [http://sourceforge.net/tracker/?group_id=75859&atid=545277] trackers on SourceForge contain descriptions of the current open issues for the Inform 7 Feature for Eclipse. If you encounter a problem, you can also search the Help forum [http://sourceforge.net/forum/forum.php?forum_id=258688] for previous reports and solutions and ask your question there.

Acknowledgements

Inform 7 [<http://www.inform-fiction.org>] was created by Graham Nelson. The text-based Linux user interface, which guided the development of the Inform 7 Feature for Eclipse, was created by Adam Thornton.

The Inform 7 Feature for Eclipse uses the public domain ANTLR 2 library developed by Terence Parr and others in the ANTLR project [<http://wwwantlr.org>].

Parts of the Inform 7 Feature's source code, design and build system were adapted from the EPIC project [<http://e-p-i-c.sf.net>].

Java is a trademark of Sun Microsystems, Inc.