The Story Editor

Michael A. Gohde

April 6, 2017

1 Overview

In order to make story authorship more accessible to a wider range of users, we have developed a graphical story editing and validation tool. This tool should make it fairly straightforward to edit stories, convert between formats, run validation tasks, and submit drafts for publication.

This document will detail the Story Editor, its features, and its usage in a production environment.

2 The Editor Window

When it is first started, the Story Editor will create a blank story and display a window similar to what is depicted above. This window has a series of controls intended to allow a user to quickly engage in common tasks, such as adding new story nodes, deleting story nodes, and updating the contents of a given story node.

As shown above, the story editor window consists of the following major components:

Story text area The story text area displays the formatted text content within the selected story node. This field also allows the user to directly edit the "source code" of a story node.

Node List This pane contains a list of all nodes currently defined in the story. Each story node is effectively an individual story unit or decision

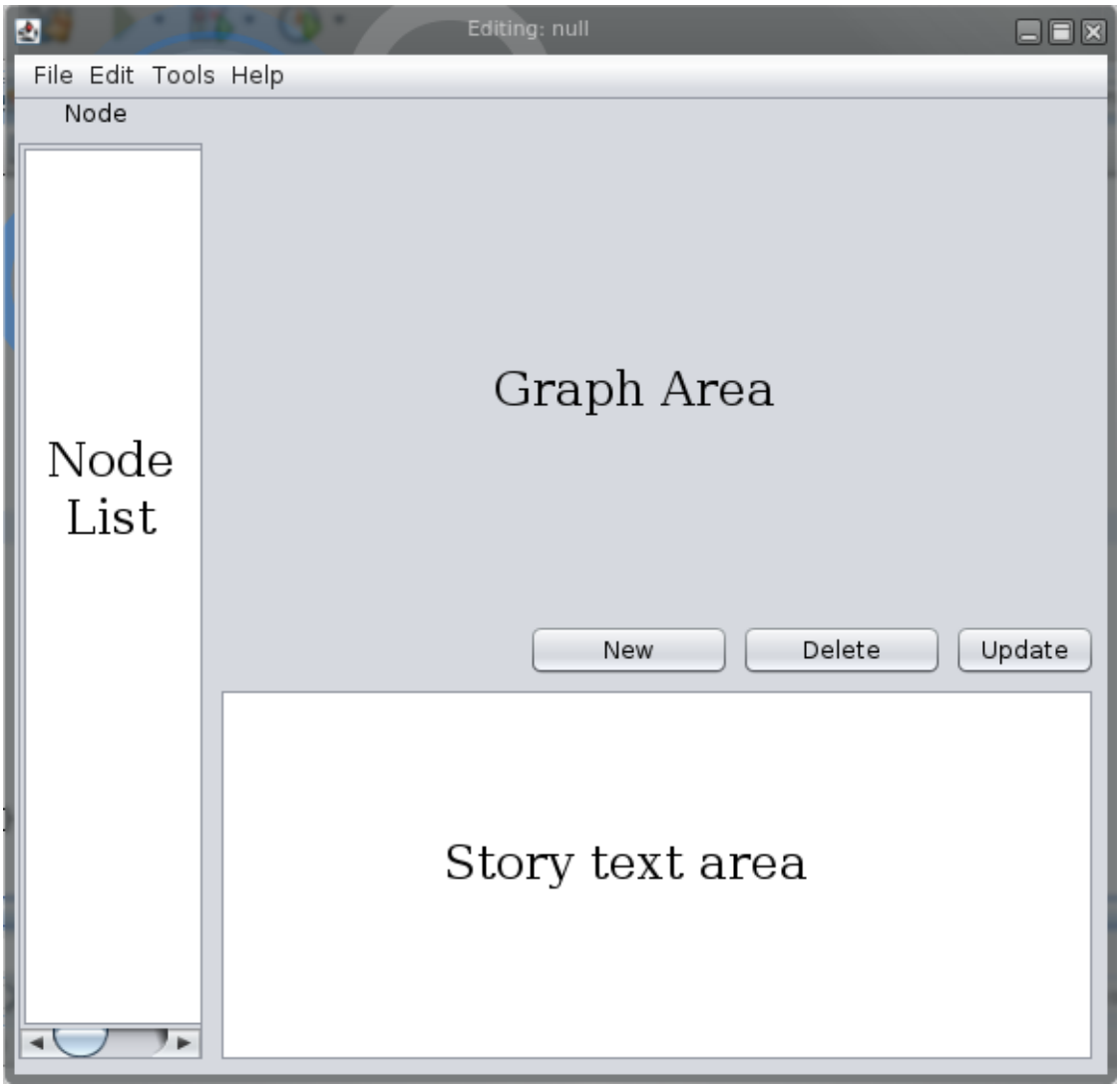

Figure 1: Elements of the editor window

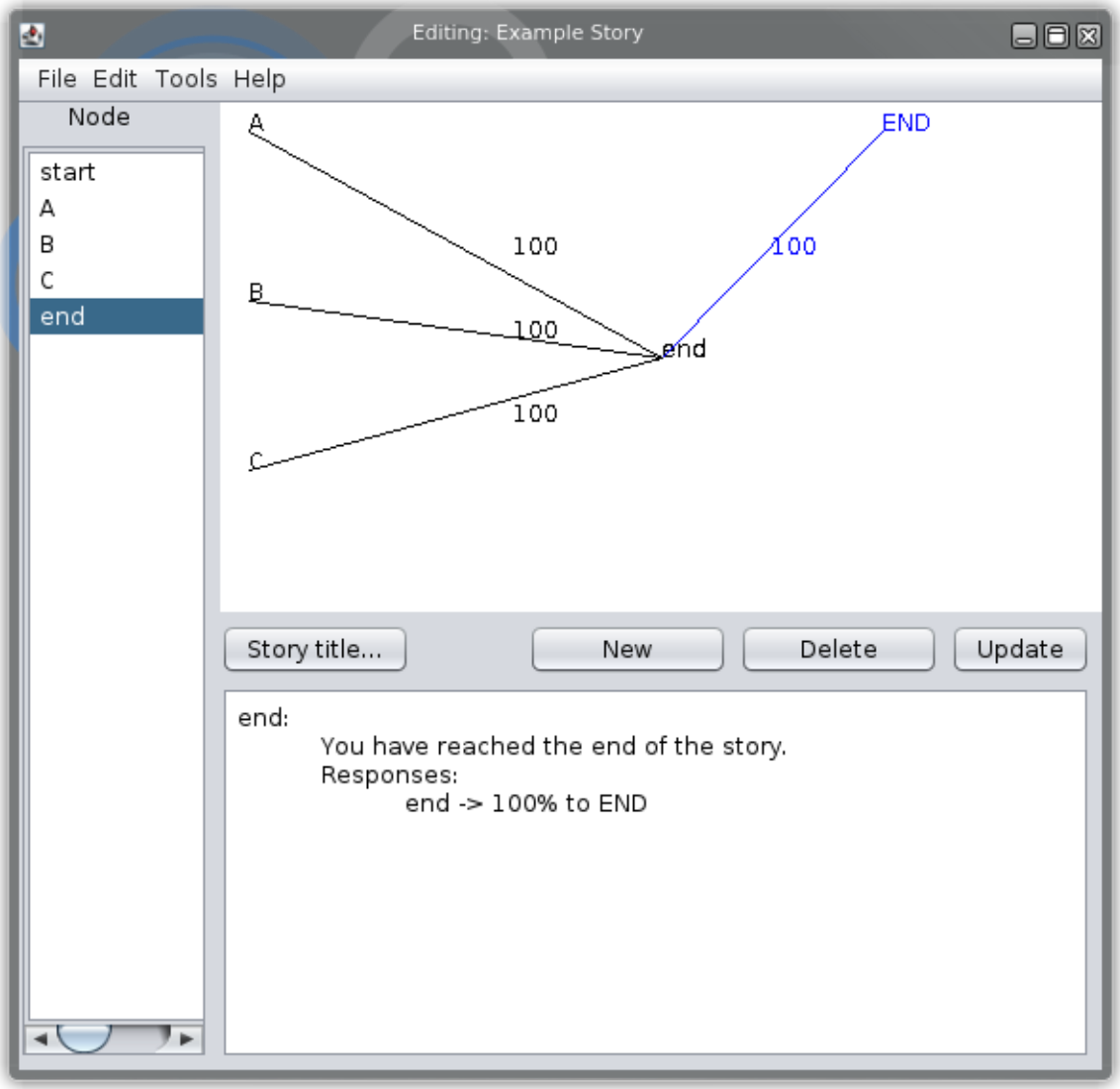

Figure 2: Editor window with active story

point that is presented to the user. The node list automatically updates when nodes are added to or removed from the story.

Graph Area The Story Editor is capable of creating a graphical representation of the currently selected node and its relationships within the story. When the Graph Area is filled, the set of all parent nodes is displayed on the right, the set of all child nodes is displayed on the left, and colored lines are drawn from the currently selected node to its parents and children.

Color	Meaning
Red	The connected node does not exist.
Blue	The connected node is an endpoint for the story.
Black	The connected node exists and is not an endpoint.

List of line colors and their meanings.

3 Menu Options

Much of the validation, import, export, and settings functionality in the Story Editor is contained in its four menus.

The File Menu The File menu contains a set of items related to creating, importing, and exporting story lines.

Item	Function
New	Creates a new story line.
Open...	Loads a file in the story definition format.
Save...	Saves a file in the story definition format.
Import from XML...	Loads a file in the XML story format used by the Nudge Support Tool set.
Export from XML...	Saves a file in the XML story format.
Import from Server...	Imports a story line from a Nudge database server.
Export to Server...	Exports a story line to a set of temporary tables in a Nudge database server.
Exit	Terminates the application.

List of items in the File menu and their functions.

The Edit Menu The Edit menu contains a set of items related to manipulating text and changing various attributes of the current story.

Item	Function
Title...	Allows the user to specify the current storyline's name.
Copy	Copies all currently selected text in the current window to the system clipboard.
. Paste	Inserts all text from the system clipboard into the current editing position.

The Tools Menu The Tools menu contains a set of items related to changing global application settings, validating the current story line, and collecting useful data.

Item	Function
Sanity test	Runs a series of tests on the current story line to determine if it is complete and sufficiently free of errors.
Publish...	Exports the story to be installed in a production Nudge server.
Set defaults...	Allows the user to specify various default values for save location, database server credentials, etc.
Create account on database server...	Creates a user account so that stories can be imported from or saved to a nudge server for easy editing elsewhere.

4 Editing a story

The story editing process is fairly straightforward.

5 Supported Story Formats

In order to maintain wide compatibility with many external tool sets and applications, the Story Editor is capable of reading and writing stories in a large number of formats. Among them, the story definition file, XML, and SQL formats provide the most utility with Nudge tools as implemented.

This section will provide examples of the same story exported to various formats.

Listing 1: Example story in story format.

```
Title: Example Story
start:
        This is the first node in the story
        Responses:
                A -> 100% to A
                B -> 50% to B, 50% to C

A:
        You selected option choice A!
        Responses:
                Proceed -> 100% to end

B:
        You chose option B and were taken to node B.
        Responses:
                Proceed -> 100% to end

C:
        You chose option B but were taken to node C!
        Responses:
                Proceed -> 100% to end

end:
        You have reached the end of the story.
        Responses:
                end -> 100% to END
```

Listing 2: Example story in XML format.

```
<story title="Example Story">
        <node id="start">
                <text>This is the first node in the
                   story</text>
                <answers>
                        <option>
                                <text>A</text>
                                <dest p="100">A</dest>
```

```
8                       </option>
9                       <option>
10                              <text>B</text>
11                              <dest p="50">B</dest>
12                              <dest p="50">C</dest>
13                      </option>
14              </answers>
15      </node>
16      <node id="A">
17              <text>You selected option choice A!</
                    text>
18              <answers>
19                      <option>
20                              <text>Proceed</text>
21                              <dest p="100">end</dest
                                    >
22                      </option>
23              </answers>
24      </node>
25      <node id="B">
26              <text>You chose option B and were taken
                     to node B.</text>
27              <answers>
28                      <option>
29                              <text>Proceed</text>
30                              <dest p="100">end</dest
                                    >
31                      </option>
32              </answers>
33      </node>
34      <node id="C">
35              <text>You chose option B but were taken
                     to node C!</text>
36              <answers>
37                      <option>
38                              <text>Proceed</text>
39                              <dest p="100">end</dest
                                    >
```

```
40                          </option>
41                  </answers>
42          </node>
43          <node id="end">
44                  <text>You have reached the end of the
                        story.</text>
45                  <answers>
46                          <option>
47                                  <text>end</text>
48                                  <dest p="100">END</dest
                                        >
49                          </option>
50                  </answers>
51          </node>
52  </story>
```

Listing 3: Set of generated SQL statements.

```
1  INSERT INTO tmpstorytable VALUES (1,'Example Story','
       start','This is the first node in the story',0);
2  INSERT INTO tmpanswers VALUES ('Example Story','start
       ','A','A');
3  INSERT INTO tmpresults VALUES (1,'Example Story','start
       ','A',0,100,'A');
4  INSERT INTO tmpanswers VALUES ('Example Story','start
       ','B','B');
5  INSERT INTO tmpresults VALUES (2,'Example Story','start
       ','B',0,50,'B');
6  INSERT INTO tmpresults VALUES (3,'Example Story','start
       ','B',50,100,'C');
7  INSERT INTO tmpstorytable VALUES (2,'Example Story','A
       ','You selected option choice A!',2);
8  INSERT INTO tmpanswers VALUES ('Example Story','A','A
       ','Proceed');
9  INSERT INTO tmpresults VALUES (4,'Example Story','A','A
       ',0,100,'end');
10 INSERT INTO tmpstorytable VALUES (3,'Example Story','B
       ','You chose option B and were taken to node B.',2);
```

```
11 INSERT INTO tmpanswers VALUES ('Example Story','B','A
    ','Proceed');
12 INSERT INTO tmpresults VALUES (5,'Example Story','B','A
    ',0,100,'end');
13 INSERT INTO tmpstorytable VALUES (4,'Example Story','C
    ','You chose option B but were taken to node C!',2);
14 INSERT INTO tmpanswers VALUES ('Example Story','C','A
    ','Proceed');
15 INSERT INTO tmpresults VALUES (6,'Example Story','C','A
    ',0,100,'end');
16 INSERT INTO tmpstorytable VALUES (5,'Example Story','
    end','You have reached the end of the story.',2);
17 INSERT INTO tmpanswers VALUES ('Example Story','end','A
    ','end');
18 INSERT INTO tmpresults VALUES (7,'Example Story','end
    ','A',0,100,'END');
```

6 Remote story editing and storage

One important possibility that the story editor enables is that of storing story lines on the a remote server so that can be more readily merged into Nudge's database. This approach should also enable users to more easily manage and collaborate on their story lines from multiple machines.

6.1 Registering for a Collaborator ID

In order to import or export story lines to a server, it is necessary to obtain two sets of credentials:

1. A database server login
2. A collaborator login

The database server login allows a user to read and write from the Nudge database itself, while the collaborator login identifies story line ownership and other key information.

In order to obtain a collaborator login, it is necessary to register with the database server. Registration is done through the "Create account on database server..." item in the "Tools" menu.

To register:

1. Open the "Create account on database server" dialog
2. Ensure that the database login credentials presented are correct.
3. Enter a desired set of collaborator credentials.
4. Click the "Check Availability..." button.
5. If the "Check Availability..." button now reads, "Register...", then click it and complete the process. Otherwise, change the desired username until it can be registered.

Once registration is complete, all relevant user authentication information will be stored automatically in the Story Editor's configuration. This means that all database interactions will automatically use the new credentials, and that the same set will be loaded on every start-up.[1]

6.2 Storing the current storyline remotely

With proper credentials, it is possible to save any active storyline to the remote server on which those credentials were created. This approach to remote storage enables a large number of new possibilities for story editing and collaboration. It can also be used to safeguard each storyline against data loss, since remote storage enables further points of redundancy.

In order to store a storyline, it is necessary to go through the following steps:

1. Click the "Export to Server..." item in the "File" menu.
2. Ensure that all information presented is correct.
3. If all information is correct, click the "Export" button.
4. A copy of the current storyline should now be stored on the remote server.

[1] It is possible to override this behavior either by using the "Set Defaults..." option in the "Tools" menu or by creating a new set of login credentials through the "Create account on database server..." item.

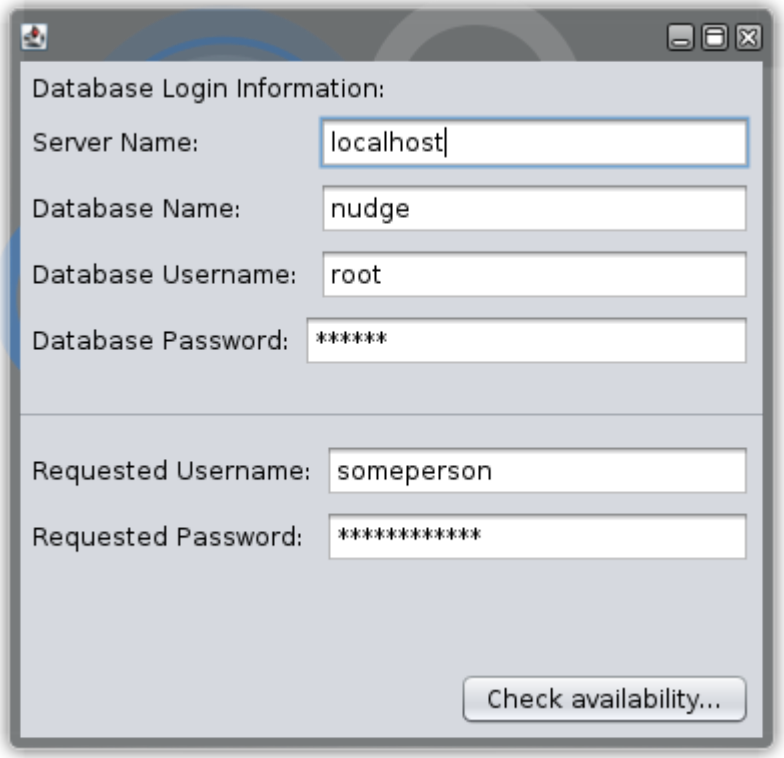

Figure 3: Collaborator registration box

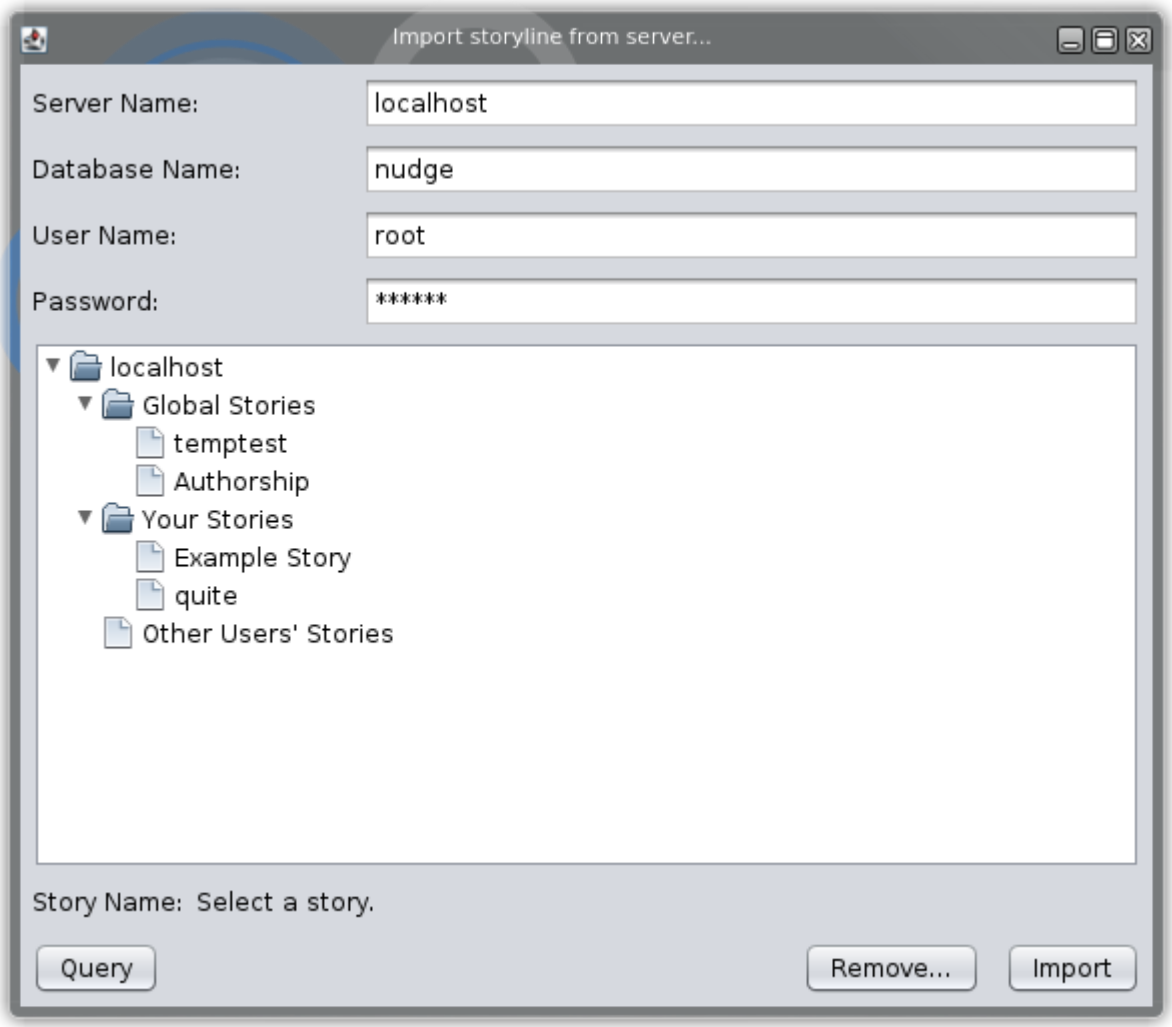

Figure 4: Story line import box

6.3 Retrieving a storyline remotely

The process of importing a storyline is very similar to that of exporting one.

7 Publishing a storyline

Once a storyline is sufficiently edited and checked, it is useful to publish it to the live Nudge server so that it can be read and interacted with by users. The publication process is fairly straightforward and requires very little additional information to be entered into the story editor's settings database.

This is the story publication process:

1. Click the "Publish" option in the "Tools" menu.
2. Ensure that the email fields present are correct.
3. If your storyline has passed automatic testing, then it should be possible to able to click the "Publish" button.
4. After "Publish" is clicked, the system mail client will start with a publication letter filled out.
5. Comment on and revise the email message as appropriate.
6. Send the message.

If all goes well, the storyline should be mailed to an authority capable of checking it over and installing it on a production Nudge server.

8 Story merge support

While not yet implemented, one important future direction for development is that of enabling users to merge story lines together. While seemingly limited in usefulness, story line merger support will enable multiple users to collaboratively edit a scoreline. This, if used correctly, has the potential to significantly reduce the time needed to write a story line.

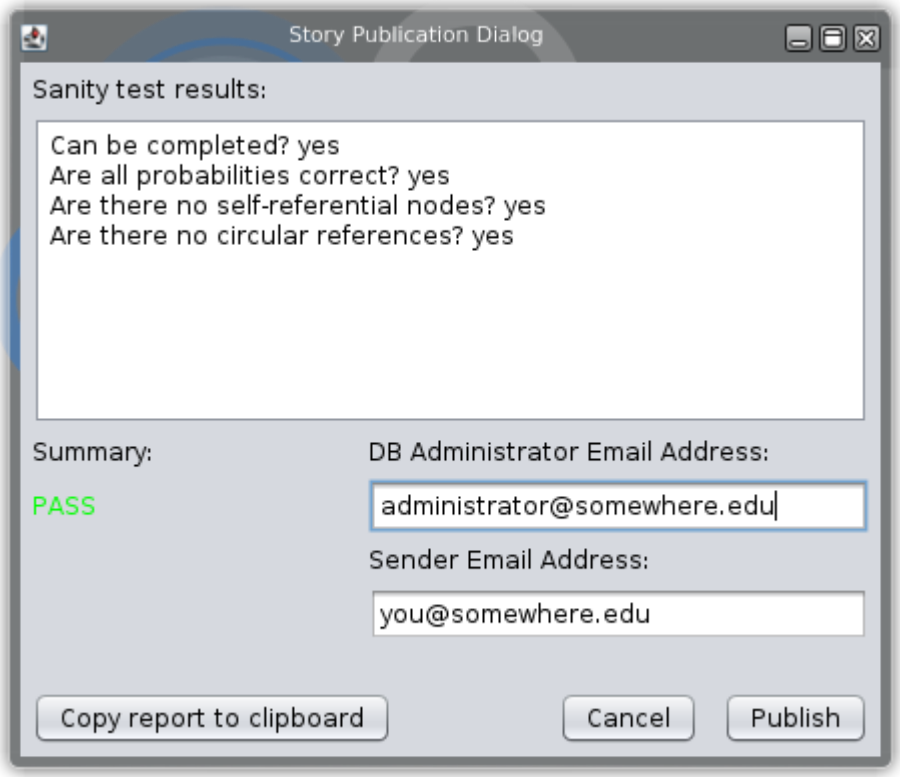

Figure 5: Story publication dialog box with valid storyline

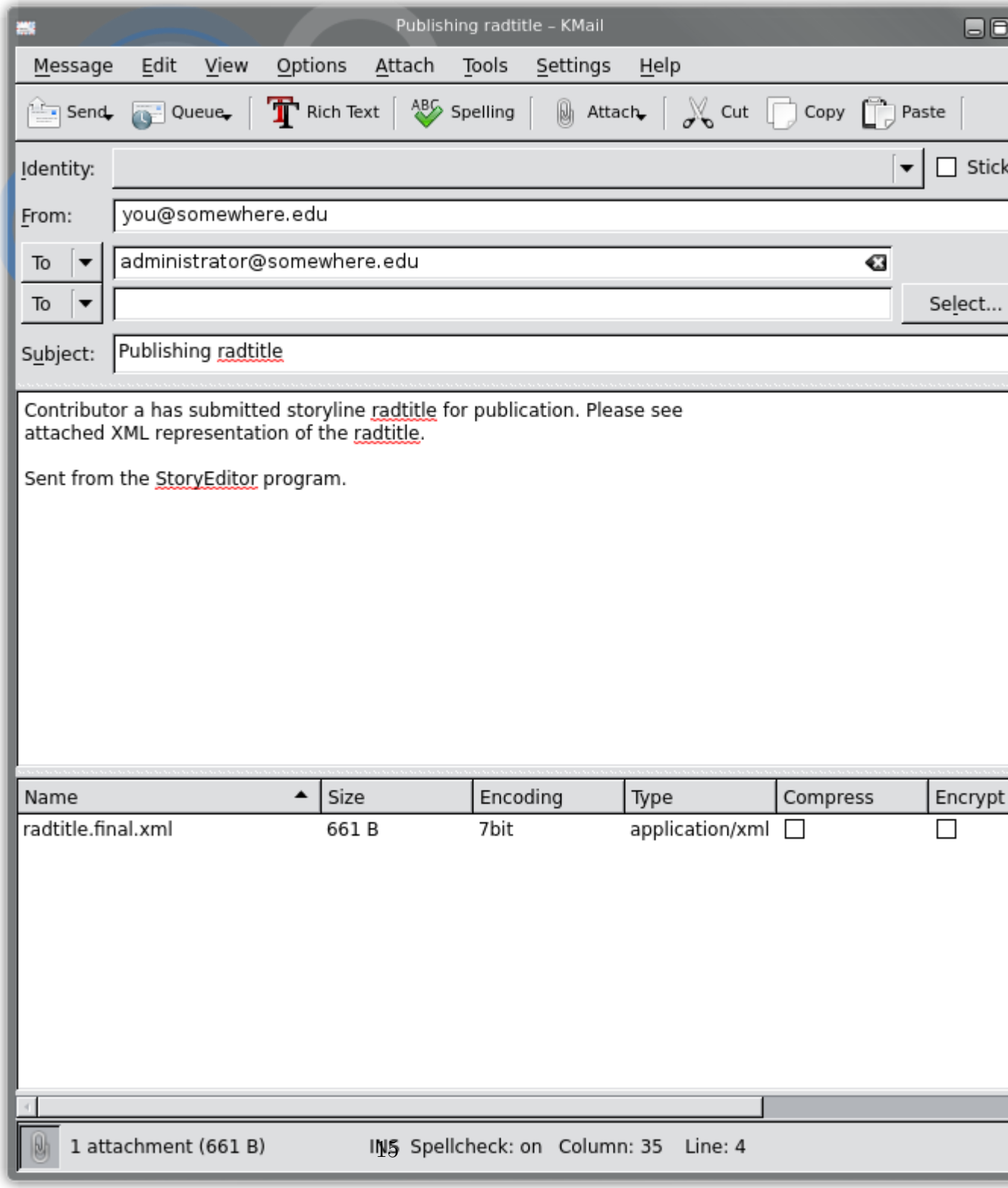

Figure 6: Unedited story email generated by the story editor