The Office of the National Coordinator for
Health Information Technology

U.S. Department of Health and Human Services (HHS)
The Office of the National Coordinator for Health Information Technology (ONC)

eConsent Trial Project

Code Set and Installation Guide for eConsent Story Engine

Produced for the U.S. Department of Health and Human Services by:
APP Design, Inc.

Version: March 2013

Contents

Figures

Acronym Index

Acronym	Definition
AAC	Advanced Audio Coding
CD	Compact Disc
DVD	Digital Versatile Disc
GB	Gigabyte
GUI	Graphical User Interface
HHS	U.S. Department of Health and Human Services
HTML	Hypertext Markup Language
MB	Megabyte
MHz	Megahertz
ONC	Office of the National Coordinator for Health Information Technology
RAM	Random Access Memory
RVM	Ruby Version Manager
USB	Universal Serial Bus
VGA	Video Graphics Array
WAV	Waveform Audio File

1.0 Introduction to Story Engine

This guide outlines the basic steps for the installation of the Story Engine application developed under this contract, which combines the Story Builder authoring tool for story creation and the Story Presenter interface to view created stories. Story Engine, with its components, provides web-based software parts that allow stories to be authored and viewed anywhere Internet connectivity is available. This guide provides instructions on how to install Story Engine on a web server.

The Story Engine code set contains two parts, Story Builder and Story Presenter. Story Builder is an authoring tool that makes it easy for authors with little technical background to create dynamic stories. Story Builder allows authors to upload files and input content to build stories — multi-page scripts consisting of Hypertext Markup Language (HTML) formatted text, images, audio, and video to create, define, and organize the navigation of the story content. Story Presenter is the interface through which patients or other viewers watch a published story created in Story Builder. Story Presenter is included as part of the installation and does not require any additional installation effort.

As an open-source tool, Story Engine can be used with a variety of operating systems, but this guide will focus on the installation procedure for the Ubuntu (Linux) operating system.[1] The installation on other Linux-based systems will be similar, and these instructions can serve as a general guide for those installations.

Because Story Engine is a server-based application, administrator privileges are required for installation of some of its components. While this software application is available to anyone, it should be noted that an intermediate level of general system administration understanding will be helpful for installation purposes. Less experienced administrators may need to seek additional knowledge elsewhere to assist them with installing Story Engine.

Additionally, because there is an inability to know server usage or server configuration of a facility installing the files discussed here, this guide does not provide an estimate on time needed to install the Story Engine files. This guide attempts to make all steps clear and concise, but some terms or operations could be unfamiliar to a less experienced administrator.

This guide covers the installation of the Story Engine only. For information about using the Story Builder authoring tool and viewing stories in Story Presenter, please review the corresponding "Graphical User Interface (GUI) Design and Authoring Guide."

[1] Story Engine can be installed on any operating system that supports the Ruby language and Rails application framework, but installation on other operating systems is outside the scope of this document. Linux is the most popular operating system for web servers, with a 65% share of publicly accessible servers. For more information, see https://en.wikipedia.org/wiki/Usage_share_of_operating_systems.

1.1 Minimum Requirements for Story Engine Installation

This section focuses on requirements as they apply to the Ubuntu (Linux) operating system. The development team selected Ubuntu because it is an open-source system, which is a requirement of the eConsent contract.

Administrators with questions about Ubuntu should review the website https://help.ubuntu.com/community/Installation for detailed information on Ubuntu installations.[2] For good performance, Ubuntu recommends administrators adhere to the following minimum criteria:

- 700 Megahertz (MHz) processor (about Intel Celeron or better)
- 512 Megabyte (MB) Random Access Memory (RAM) (system memory)
- 5 Gigabytes (GBs) of hard drive space
- Video Graphics Array (VGA) capable of 1024x768 screen resolution
- Either a Compact Disc (CD)/Digital Versatile Disc (DVD) drive or a Universal Serial Bus (USB) port for the installer media
- Internet access (required)

The requirements for other Linux-based systems should be similar, though recommendations or suggestions regarding other operating systems cannot be made because other systems have not been tested for this application.

1.2 Story Engine (Builder & Presenter)

Story Engine is a Ruby on Rails (version 3.2) application which creates presentations by accessing files maintained in different stories/modules and utilizing them in selected story lines. On the authoring side, by holding files in separate modules, Story Builder permits a single file to be used concurrently in several different stories. The modular system allows several stories to be generated from the same pool of files without needing to draft new files for every story production. Story Builder supports authoring question script, answers, alternatives, and control logic about the desired topic. Review the corresponding “GUI Design and Authoring Guide” for more information about authoring use.

1.3 Ruby and Ruby on Rails

Ruby is an open-source, dynamic, object-oriented scripting language. Ruby on Rails, often referred to as “Rails,” is an open source web application framework for the Ruby programming language. Ruby and Ruby on Rails are separate software components. This document refers to the programming language as “Ruby” and the application framework as “Rails.”

1.4 Ruby Version Manager (RVM)

Ruby Version Manager (RVM) is a utility for installing and managing multiple versions of the Ruby language without affecting existing Ruby installations. RVM also manages Ruby libraries, known as “gems,” allowing multiple Ruby/Rails application framework installations to use

[2] The recommended minimum system requirements, according to Ubuntu’s website, allow “even someone fairly new to installing Ubuntu to easily install a usable system with enough room to be comfortable.”

different gems without conflicts. RVM will be used during this Story Engine installation process to establish the Ruby instance and gems without altering existing system settings.

1.5 Open Source Packages and Repositories

The Ruby and Rails open source packages are downloaded from Ubuntu source repositories. The repositories used depend on the server configuration and the Ubuntu version. The sources used for a specific server are listed in /etc/apt/sources.list. Here is an example list:

```
# sources.list
# deb cdrom:[Ubuntu 8.04.1 _Hardy Heron_ - Release amd64 (20080701)]/ hardy main
restricted

#deb cdrom:[Ubuntu 8.04.1 _Hardy Heron_ - Release amd64 (20080701)]/ hardy main
restricted # See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to #
newer versions of the distribution.

deb http://us.archive.ubuntu.com/ubuntu/ hardy main restricted deb-src
http://us.archive.ubuntu.com/ubuntu/ hardy main restricted

## Major bug fix updates produced after the final release of the ## distribution.
deb http://us.archive.ubuntu.com/ubuntu/ hardy-updates main restricted deb-src
http://us.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
```

The size of the packages downloaded is dependent on the server configuration and Ubuntu version. Package sizes (compressed) for the example install are:

- RVM: .3MB
- Ruby/Libraries: 40MB
- Story Engine: .5MB
- Story Engine Libraries: 50MB

The Ubuntu community verifies their source repositories, which are likely (but not guaranteed) to be free of viruses. If desired, open source packages can be installed via “apt-secure” (instead of “apt-get”) which uses digital signatures to verify the packages.

2.0 Installation of Components

This Story Engine installation requires that the four main software components be installed in the following order:

1. RVM
2. Ruby
3. Rails
4. Story Engine (including Story Builder and Story Presenter)

2.1 Installing Ruby and the Rails Application Framework

2.1.1 Installation Directory

During the initial installation, a facility may do a beta installation to a local (home) directory because the installation is a work in progress as authors learn the application and testing and training take place. For a production installation (and depending on the system administrator guidelines), the installation might go to a different location. For a large environment of, for example, 25 to 50 authors, the system administrator installing Story Engine might have a specific installation directory. Large environment installations typically depend on system administrator guidelines and are beyond the scope of this document.

For example, the eConsent pilot technology platform is considered a small-to-medium-scale environment due to the relatively small number of authors and users involved. These types of environments are typically installed in either the installer's home directory or the "/opt" directory. Please consult your system administrator for the recommended directory if not installing in your home directory.

2.1.2 Command Line Interface

The installation is performed from the command line interface. Below are several ways an administrator can access the command line; methods of launching can vary by desktop environment.

- **Unity (Ubuntu 11.04)**
  Dash -> Search for Terminal
  Dash -> More Apps -> 'See More Results' -> Terminal
  Dash -> More Apps -> Accessories -> Terminal
- **GNOME Desktop (Classic Ubuntu)**
  Applications | Accessories | Terminal
- **KDE Desktop (Kubuntu)**
  KDE Menu | Applications | System | Konsole Terminal Program
- **XFCE Desktop (Xubuntu)**
  Applications | System | Terminal

Since the administrator is mostly interacting through the keyboard once he/she is on the command line, here are alternative ways to invoke the command line through the keyboard:

- **Unity (Ubuntu)**
  Ctl + Alt + T
- **GNOME (Ubuntu)**
  Alt + F2 -> (Type within the text box) gnome-terminal (Press return)
- **KDE (Kubuntu)**
  Alt + F2 -> (Type within the text box) konsole (Press return)

2.2 Installing Ruby Version Manager (RVM)

RVM must be installed on the system prior to any Ruby components and requires the installation of the following three software components in the following order:

1. curl – a web data transfer tool
2. git – a source code control manager
3. build-essential – a package of tools used to build Ruby and Ruby libraries

Curl, git, and build-essential are retrieved from the standard Ubuntu repositories as part of the apt-get command. RVM is different and is retrieved from its repository at http://rvm.io.

Following are the steps for installing RVM:

1. Install curl and git using the command *sudo apt-get install git-core curl.*
2. Install build-essential using the command *sudo apt-get install build-essential.*
3. Install RVM using the command *curl -L get.rvm.io | bash -s stable.* The output of the command should look like this:

```
% Total % Received % Xferd Average Speed Time Time Time Current
 Dload Upload Total Spent Left Speed
100 185 100 185 0 0 223 0 --:--:-- --:--:-- --:--:-- 1516
100 9439 100 9439 0 0 6212 0 0:00:01 0:00:01 --:--:-- 6212
Downloading RVM from wayneeseguin branch stable
 % Total % Received % Xferd Average Speed Time Time Time Current
 Dload Upload Total Spent Left Speed
100 125 100 125 0 0 167 0 --:--:-- --:--:-- --:--:-- 324
100 1046k 100 1046k 0 0 243k 0 0:00:04 0:00:04 --:--:-- 332k

Installing RVM to /home/econsent/.rvm/
 Adding rvm PATH line to /home/econsent/.bashrc /home/econsent/.zshrc.
 Adding rvm loading line to /home/econsent/.bash_login /home/econsent/.zlogin.

# RVM: Shell scripts enabling management of multiple ruby environments.
# RTFM: https://rvm.io/
# HELP: http://webchat.freenode.net/?channels=rvm (#rvm on irc.freenode.net)
# Cheatsheet: http://cheat.errtheblog.com/s/rvm/
# Screencast: http://screencasts.org/episodes/how-to-use-rvm

# In case of any issues read output of 'rvm requirements' and/or 'rvm notes'

Installation of RVM in /home/econsent/.rvm/ is almost complete:

 * To start using RVM you need to run `source /home/econsent/.rvm/scripts/rvm`
 in all your open shell windows, in rare cases you need to reopen all shell windows.

# econsent,
#
# Thank you for using RVM!
# I sincerely hope that RVM helps to make your life easier and more enjoyable!!!
#
# ~Wayne

rvm 1.14.5 (stable) by Wayne E. Seguin <wayneeseguin@gmail.com>, Michal Papis
<mpapis@gmail.com> [https://rvm.io/]
```

   This installs RVM to the current home directory and provides a development environment separate from the rest of the system.

4. Run /home/econsent/.rvm/scripts/rvm to start RVM. This command should only need to be run once. RVM should be started on shell start from then on.
5. The previous command should have started the RVM. Verify that the RVM is running by entering the command *rvm –v*. The output of the command should look like this:

```
% rvm 1.14.5 (stable) by Wayne E. Seguin <wayneeseguin@gmail.com>, Michal Papis
<mpapis@gmail.com> [https://rvm.io/]
```

2.3 Installing Ruby Programming Language

Before beginning the Ruby installation, you must identify and retrieve the software gems required by Ruby.

1. Use the *rvm requirements* command to identify the required Ruby gems. The output of the command should look similar to this:

```
# For Ruby / Ruby HEAD (MRI, Rubinius, & REE), install the following:
 ruby: /usr/bin/apt-get install build-essential openssl libreadline6
 libreadline6-dev curl git-core zlib1g zlib1g-dev libssl-dev libyaml-dev
 libsqlite3-0 libsqlite3-dev sqlite3 libxml2-dev libxslt-dev autoconf libc6-dev
ncurses-dev automake libtool bison subversion
```

2. The Ruby gems returned by this command must be downloaded to your system via the Internet. This may take several minutes depending upon the speed of your Internet connection. Use the following *apt-get* command to download the gems to your system. You will be prompted for permission before the download begins.

   sudo apt-get install build-essential openssl libreadline6 libreadline6-dev curl git-core zlib1g zlib1g-dev libssl-dev libyaml-dev libsqlite3-0 libsqlite3-dev sqlite3 libxml2-dev libxslt-dev autoconf libc6-dev ncurses-dev automake libtool bison subversion libxml2 libxml2-devel libxslt libxslt-devel

3. Install Ruby with the command *rvm install 1.9.3*. The output of the command should look similar to this:

```
Ruby (and needed base gems) for your selection will be installed shortly.
Before it happens, please read and execute the instructions below.
Please use a separate terminal to execute any additional commands.

Requirements for Linux ( DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=11.10
DISTRIB_CODENAME=oneiric
DISTRIB_DESCRIPTION="Ubuntu 11.10" )

NOTE: 'ruby' represents Matz's Ruby Interpreter (MRI) (1.8.X, 1.9.X)
 This is the *original* / standard Ruby Language Interpreter
 'ree' represents Ruby Enterprise Edition
 'rbx' represents Rubinius

bash >= 4.1 required
curl is required
git is required (>= 1.7 for ruby-head)
patch is required (for 1.8 rubies and some ruby-head's).

To install rbx and/or Ruby 1.9 head (MRI) (eg. 1.9.2-head),
then you must install and use rvm 1.8.7 first.

Additional Dependencies:
# For Ruby / Ruby HEAD (MRI, Rubinius, & REE), install the following:
 ruby: /usr/bin/apt-get install build-essential openssl libreadline6 libreadline6-dev
curl git-core zlib1g zlib1g-dev libssl-dev libyaml-dev libsqlite3-dev sqlite3 libxml2-
dev libxslt-dev autoconf libc6-dev ncurses-dev automake libtool bison subversion
```

```
# For JRuby, install the following:
 jruby: /usr/bin/apt-get install curl g++ openjdk-6-jre-headless
 jruby-head: /usr/bin/apt-get install ant openjdk-6-jdk

# For IronRuby, install the following:
 ironruby: /usr/bin/apt-get install curl mono-2.0-devel

Press 'q' to continue.
Fetching yaml-0.1.4.tar.gz to /home/econsent/.rvm/archives
 % Total % Received % Xferd Average Speed Time Time Time Current
 Dload Upload Total Spent Left Speed
100 460k 100 460k 0 0 172k 0 0:00:02 0:00:02 --:--:-- 208k
Extracting yaml-0.1.4.tar.gz to /home/econsent/.rvm/src
Prepare yaml in /home/econsent/.rvm/src/yaml-0.1.4.
Configuring yaml in /home/econsent/.rvm/src/yaml-0.1.4.
Compiling yaml in /home/econsent/.rvm/src/yaml-0.1.4.
Installing yaml to /home/econsent/.rvm/usr
Installing Ruby from source to: /home/econsent/.rvm/rubies/ruby-1.9.3-p194, this may
take a while depending on your cpu(s)...

ruby-1.9.3-p194 - #fetching
ruby-1.9.3-p194 - #downloading ruby-1.9.3-p194, this may take a while depending on
your connection...
 % Total % Received % Xferd Average Speed Time Time Time Current
 Dload Upload Total Spent Left Speed
100 9610k 100 9610k 0 0 55483 0 0:02:57 0:02:57 --:--:-- 45420
ruby-1.9.3-p194 - #extracting ruby-1.9.3-p194 to /home/econsent/.rvm/src/ruby-1.9.3-
p194
ruby-1.9.3-p194 - #extracted to /home/econsent/.rvm/src/ruby-1.9.3-p194
ruby-1.9.3-p194 - #configuring
ruby-1.9.3-p194 - #compiling
ruby-1.9.3-p194 - #installing
Retrieving rubygems-1.8.24
 % Total % Received % Xferd Average Speed Time Time Time Current
 Dload Upload Total Spent Left Speed
100 371k 100 371k 0 0 330k 0 0:00:01 0:00:01 --:--:-- 463k
Extracting rubygems-1.8.24 ...
Removing old Rubygems files...
Installing rubygems-1.8.24 for ruby-1.9.3-p194 ...
Installation of rubygems completed successfully.
ruby-1.9.3-p194 - adjusting #shebangs for (gem irb erb ri rdoc testrb rake).
ruby-1.9.3-p194 - #importing default gemsets (/home/econsent/.rvm/gemsets/)
Install of ruby-1.9.3-p194 - #complete
```

4. When the installation process has completed, enter the command *rvm list* to verify that Ruby is installed. The output of the command should look similar to this:

```
rvm rubies

=* ruby-1.9.3-p194 [ i686 ]

# => - current
# =* - current && default
# * - default
```

5. Verify you are using the newly installed version of Ruby by entering the command *ruby –v.* The output of the command should look similar to this (you may see a larger number after the "p," indicating a more recent revision of Ruby):

```
ruby 1.9.3p194 (2012-04-20 revision 35410) [i686-linux]
```

6. Set this as the default version of Ruby using the command *rvm --default use 1.9.3-p194*. The output of the command should look similar to this:

```
Using /home/econsent/.rvm/gems/ruby-1.9.3-p194
```

2.4 Installing Rails Application Framework

Install the Rails gem using the command *gem install rails -v 3.2.3*. The output should look like this:

```
Fetching: i18n-0.6.0.gem (100%)
Fetching: multi_json-1.3.6.gem (100%)
Fetching: activesupport-3.2.3.gem (100%)
Fetching: builder-3.0.0.gem (100%)
Fetching: activemodel-3.2.3.gem (100%)
Fetching: rack-1.4.1.gem (100%)
Fetching: rack-cache-1.2.gem (100%)
Fetching: rack-test-0.6.1.gem (100%)
Fetching: journey-1.0.4.gem (100%)
Fetching: hike-1.2.1.gem (100%)
Fetching: tilt-1.3.3.gem (100%)
Fetching: sprockets-2.1.3.gem (100%)
Fetching: erubis-2.7.0.gem (100%)
Fetching: actionpack-3.2.3.gem (100%)
Fetching: arel-3.0.2.gem (100%)
Fetching: tzinfo-0.3.33.gem (100%)
Fetching: activerecord-3.2.3.gem (100%)
Fetching: activeresource-3.2.3.gem (100%)
Fetching: mime-types-1.19.gem (100%)
Fetching: polyglot-0.3.3.gem (100%)
Fetching: treetop-1.4.10.gem (100%)
Fetching: mail-2.4.4.gem (100%)
Fetching: actionmailer-3.2.3.gem (100%)
Fetching: thor-0.14.6.gem (100%)
Fetching: rack-ssl-1.3.2.gem (100%)
Fetching: json-1.7.3.gem (100%)
Building native extensions. This could take a while...
Fetching: rdoc-3.12.gem (100%)
Depending on your version of ruby, you may need to install ruby rdoc/ri data:

<= 1.8.6 : unsupported
 = 1.8.7 : gem install rdoc-data; rdoc-data --install
 = 1.9.1 : gem install rdoc-data; rdoc-data --install
>= 1.9.2 : nothing to do! Yay!
Fetching: railties-3.2.3.gem (100%)
Fetching: rails-3.2.3.gem (100%)
Successfully installed i18n-0.6.0
Successfully installed multi_json-1.3.6
Successfully installed activesupport-3.2.3
Successfully installed builder-3.0.0
Successfully installed activemodel-3.2.3
Successfully installed rack-1.4.1
Successfully installed rack-cache-1.2
Successfully installed rack-test-0.6.1
Successfully installed journey-1.0.4
Successfully installed hike-1.2.1
Successfully installed tilt-1.3.3
Successfully installed sprockets-2.1.3
Successfully installed erubis-2.7.0
Successfully installed actionpack-3.2.3
Successfully installed arel-3.0.2
Successfully installed tzinfo-0.3.33
Successfully installed activerecord-3.2.3
Successfully installed activeresource-3.2.3
Successfully installed mime-types-1.19
Successfully installed polyglot-0.3.3
Successfully installed treetop-1.4.10
Successfully installed mail-2.4.4
Successfully installed actionmailer-3.2.3
Successfully installed thor-0.14.6
Successfully installed rack-ssl-1.3.2
Successfully installed json-1.7.3
Successfully installed rdoc-3.12
Successfully installed railties-3.2.3
Successfully installed rails-3.2.3
29 gems installed
Installing ri documentation for i18n-0.6.0...
Installing ri documentation for multi_json-1.3.6...
Installing ri documentation for activesupport-3.2.3...
Installing ri documentation for builder-3.0.0...
Installing ri documentation for activemodel-3.2.3...
Installing ri documentation for rack-1.4.1...
Installing ri documentation for rack-cache-1.2...
Installing ri documentation for rack-test-0.6.1...
Installing ri documentation for journey-1.0.4...
Installing ri documentation for hike-1.2.1...
Installing ri documentation for tilt-1.3.3...
Installing ri documentation for sprockets-2.1.3...
Installing ri documentation for erubis-2.7.0...
Installing ri documentation for actionpack-3.2.3...
Installing ri documentation for arel-3.0.2...
Installing ri documentation for tzinfo-0.3.33...
Installing ri documentation for activerecord-3.2.3...
Installing ri documentation for activeresource-3.2.3...
Installing ri documentation for mime-types-1.19...
Installing ri documentation for polyglot-0.3.3...
Installing ri documentation for treetop-1.4.10...
Installing ri documentation for mail-2.4.4...
Installing ri documentation for actionmailer-3.2.3...
Installing ri documentation for thor-0.14.6...
Installing ri documentation for rack-ssl-1.3.2...
Installing ri documentation for json-1.7.3...
Installing ri documentation for rdoc-3.12...
Installing ri documentation for railties-3.2.3...
Installing ri documentation for rails-3.2.3...
Installing RDoc documentation for i18n-0.6.0...
Installing RDoc documentation for multi_json-1.3.6...
Installing RDoc documentation for activesupport-3.2.3...
Installing RDoc documentation for builder-3.0.0...
Installing RDoc documentation for activemodel-3.2.3...
Installing RDoc documentation for rack-1.4.1...
Installing RDoc documentation for rack-cache-1.2...
Installing RDoc documentation for rack-test-0.6.1...
Installing RDoc documentation for journey-1.0.4...
Installing RDoc documentation for hike-1.2.1...
Installing RDoc documentation for tilt-1.3.3...
Installing RDoc documentation for sprockets-2.1.3...
Installing RDoc documentation for erubis-2.7.0...
Installing RDoc documentation for actionpack-3.2.3...
Installing RDoc documentation for arel-3.0.2...
Installing RDoc documentation for tzinfo-0.3.33...
Installing RDoc documentation for activerecord-3.2.3...
Installing RDoc documentation for activeresource-3.2.3...
Installing RDoc documentation for mime-types-1.19...
Installing RDoc documentation for polyglot-0.3.3...
Installing RDoc documentation for treetop-1.4.10...
Installing RDoc documentation for mail-2.4.4...
Installing RDoc documentation for actionmailer-3.2.3...
Installing RDoc documentation for thor-0.14.6...
Installing RDoc documentation for rack-ssl-1.3.2...
Installing RDoc documentation for json-1.7.3...
Installing RDoc documentation for rdoc-3.12...
Installing RDoc documentation for railties-3.2.3...
Installing RDoc documentation for rails-3.2.3...
```

This command will install the Rails gem and any dependent libraries, including bundler, which is required for the Story Engine install. Note that the Ubuntu sudo utility is not being used here as we only want to install Rails locally without altering other existing system settings.

The Sqlite3 database which is part of the Rails Application Framework is being used, so there is no need to install a separate database or driver.

2.5 Installing Story Engine/Establishing Default Administrator

The steps for installing Story Engine are as follows:

1. Download the latest Story Engine distribution file from GitHub at https://github.com.
2. Unzip the downloaded distribution file in your home directory.
3. Change to the Story Engine distribution directory using the command *cd ~/storyengine*.
4. Run the bundle utility using the command *bundle install*. The output should look like this:

```
Fetching gem metadata from https://rubygems.org/.......
Using rake (0.9.2.2)
Using i18n (0.6.0)
Installing multi_json (1.2.0)
Using activesupport (3.2.3)
Using builder (3.0.0)
Using activemodel (3.2.3)
Using erubis (2.7.0)
Installing journey (1.0.3)
Using rack (1.4.1)
Using rack-cache (1.2)
Using rack-test (0.6.1)
Using hike (1.2.1)
Using tilt (1.3.3)
Installing sprockets (2.1.2)
Using actionpack (3.2.3)
Installing mime-types (1.18)
Using polyglot (0.3.3)
Using treetop (1.4.10)
Using mail (2.4.4)
Using actionmailer (3.2.3)
Using arel (3.0.2)
Using tzinfo (0.3.33)
Using activerecord (3.2.3)
Using activeresource (3.2.3)
Installing acts_as_list (0.1.6)
Installing bcrypt-ruby (3.0.1) with native extensions
Installing cocaine (0.2.1)
Installing coderay (1.0.6)
Installing coffee-script-source (1.2.0)
Installing execjs (1.3.0)
Installing coffee-script (2.2.0)
Using rack-ssl (1.3.2)
Installing json (1.6.6) with native extensions
Using rdoc (3.12)
Using thor (0.14.6)
Using railties (3.2.3)
Installing coffee-rails (3.2.2)
Installing commonjs (0.2.5)
Installing daemons (1.1.8)
Installing orm_adapter (0.0.7)
Installing warden (1.1.1)
Installing devise (2.0.4)
Installing diff-lcs (1.1.3)
Installing eventmachine (0.12.10) with native extensions
Installing jquery-rails (2.0.2)
Installing libv8 (3.3.10.4)
Installing therubyracer (0.10.1) with native extensions
Installing less (2.1.0)
Installing less-rails (2.2.0)
Installing method_source (0.7.1)
Installing paperclip (3.0.2)
Installing slop (2.4.4)
Installing pry (0.9.8.4)
Using bundler (1.1.4)
Using rails (3.2.3)
Installing rspec-core (2.9.0)
Installing rspec-expectations (2.9.1)
Installing rspec-mocks (2.9.0)
Installing rspec (2.9.0)
Installing rspec-rails (2.9.0)
Installing sass (3.1.15)
Installing sass-rails (3.2.5)
Installing spork (0.9.2)
Installing sqlite3 (1.3.5) with native extensions
Installing thin (1.3.1) with native extensions
Installing tinymce-rails (3.4.9)
Installing twitter-bootstrap-rails (2.0.6)
Installing uglifier (1.2.4)
Your bundle is complete! Use `bundle show [gemname]` to see where a bundled gem is
installed.
```

This command will retrieve and install all of the gems needed by Story Engine.

5. Run the Rake task *rake db:schema:load* to create the database.
6. Run the Rake task *rake db:seed* **to set up the initial administrator user.** The email address for this initial user will be admin@appdesign.com, and the password will be "econsentadmin" (all lowercase). This password should be changed after the first login. Review Section 3.3 Administrative Password for details on changing the password.
7. Once the bundler utility finishes, start Rails using the command *rails server*.
   The command line output should look like this:

```
=> Booting Thin
=> Rails 3.2.3 application starting in development on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
>> Thin web server (v1.3.1 codename Triple Espresso)
>> Maximum connections set to 1024
>> Listening on 0.0.0.0:3000, CTRL+C to stop
```

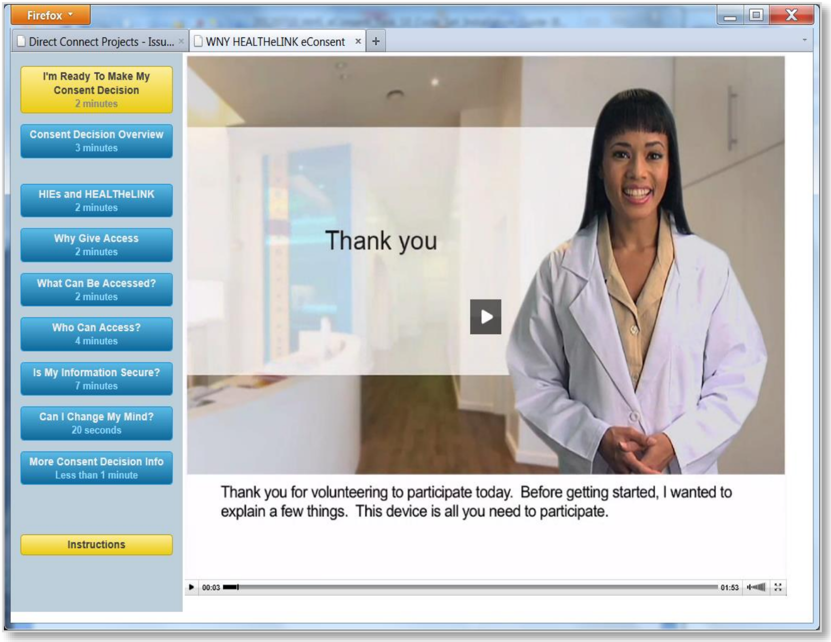
8. You should be able to open a web browser at http://localhost:3000 and see the ONC/eConsent introduction page.
9. To log in to Story Engine, go to http://localhost:3000/login. Use the login and password given in step 6 above.

Refer to “GUI Design and Authoring Guide” for information on setting up administrative and general user accounts.

3.0 Additional Technical Considerations

3.1 Story Presenter

Story Presenter’s interface is a window/interface for viewing content created in Story Builder (see Figure 1 below). Story Presenter reads the story scripts generated by Story Builder and sends them to the web browser for display. Story Presenter is part of the Story Engine installation; a separate install is not required.


Firefox
Direct Connect Projects - Issu...
WNY HEALTHeLINK eConsent
I'm Ready To Make My Consent Decision 2 minutes
Consent Decision Overview 3 minutes
HIEs and HEALTHeLINK 2 minutes
Why Give Access 2 minutes
What Can Be Accessed? 2 minutes
Who Can Access? 4 minutes
Is My Information Secure? 7 minutes
Can I Change My Mind? 20 seconds
More Consent Decision Info Less than 1 minute
Instructions
Thank you
00:03
01:53

Thank you for volunteering to participate today. Before getting started, I wanted to explain a few things. This device is all you need to participate.

Figure 1: Example Interface in Story Presenter

There are no external controls for Story Presenter other than a user’s mouse if he/she uses a desktop or laptop computer. If a touchscreen-enabled device is used, then a mouse is not needed because users touch the virtual, on-screen buttons (designed in Story Builder) to review each page or to select an answer to a question.

3.2 Web Browser Compatibility

Story Presenter uses a web browser to display the story content. Story Presenter is compatible with the web browsers and operating systems listed below. For more information about audio/video compatibility, see Section 3.5.

Browser	Operating Systems		
	Windows XP	Mac OSX	GNU/Linux
Internet Explorer 10	OK	N/A	N/A
Internet Explorer 9	OK	N/A	N/A
Internet Explorer 8	OK	N/A	N/A
Internet Explorer 7	OK	N/A	N/A
Internet Explorer 6	OK	N/A	N/A
Firefox	OK	OK	OK
Chrome	OK	OK	OK
Safari	OK	OK	N/A
Opera	OK	OK	OK

Figure 2: Web Browser Capability

3.3 Administrative Password

After the Story Engine application has been installed, the administrator should change the default password that was created during the steps discussed in Section 2.5. Story Engine must be installed first before the password can be changed. An administrative password (econsentadmin) and administrative user ID (admin@appdesign.com) are hard-coded into the installation process for initial setup.

To change the administrative password (referred to as "password" moving forward in this document), the administrator must log into Story Builder using the default user ID and password noted above. The authors recommend this be done immediately upon logging in the first time to Story Builder.

After logging in, the following screen should display:

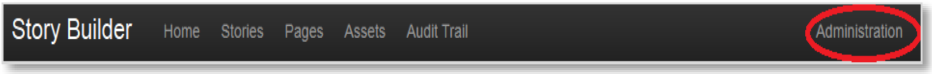
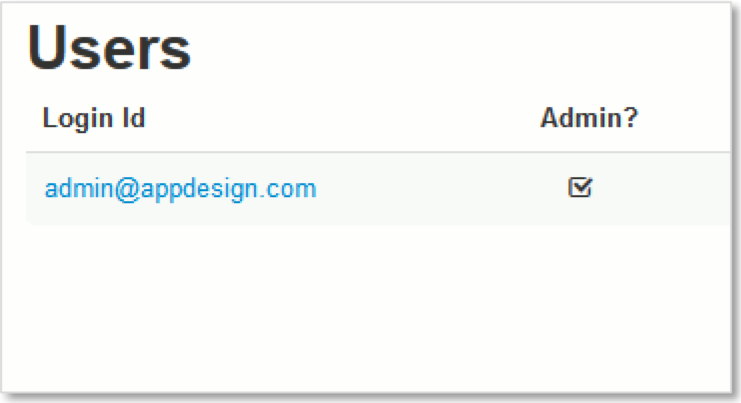


Figure 3: Story Builder Menu Display Upon Login

1. Press the Administration link on the right hand corner of the menu. The administrator will be taken to the Users page (see Figure 4 on next page). Note the admin designation boxes. If a user has this box checked, he/she will be able to view/modify/delete users. The default administrator user (admin@appdesign.com) will be checked automatically.

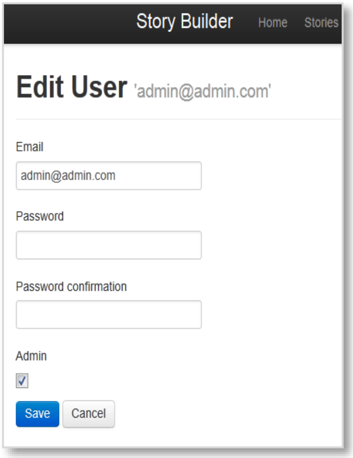


Users

Login Id	Admin?
admin@appdesign.com	☑

Figure 4: Users Page

2. Press the admin@appdesign.com link. The administrator will be taken to the Edit User form (see Figure 5 below).



Edit User 'admin@admin.com'

Email: admin@admin.com

Password:

Password confirmation:

Admin: ☑

Figure 5: Edit User Form

3. Type the new password into the Password and Password Confirmation fields. Be sure the "Admin" box is still checked.
4. Press the "Save" button to capture the change.

Note: Anyone with administrative privileges has full access to view/modify/delete users and passwords. Only trusted users should be given administrative privileges in Story Builder.

Refer to the "GUI Design and Authoring Guide" for information on setting up additional administrative and general user accounts.

3.4 Video and Audio Player

To play video and audio content, Story Presenter uses the open source Video.js video and audio player. This player is invoked automatically to play video and audio assets on the server on which the Story Presenter is installed. The author only has to specify the desired asset to be played in Story Builder, and Story Presenter will play it

For more information on the video player, see http://videojs.com/.

3.5 Audio/Video Compatibility

The open source audio/video player included with Story Presenter will play standard MP4 video files.[3] If the web browser does not support native video playback, it will fall back to a built-in Adobe Flash player. The open source player has been successfully tested to play MP4-formatted files on Firefox 18.0, Internet Explorer 9, and Google Chrome 24.0 browsers.

Internet Explorer only supports MP3 and Advanced Audio Coding (AAC) audio formats. Firefox supports Ogg Vorbis and Waveform Audio File (WAV) formats. Chrome supports Ogg Vorbis and MP3 formats. If the web browser does not support the chosen audio format, the open source audio/video player can NOT fall back to the built-in Adobe Flash player for audio.

[3] Care should be taken to choose the proper audio format to meet your needs, because not all web browsers support or are compatible with all audio and video formats playable by the open source player. For the eConsent pilot, the MP4 video format and WAV audio formats were used. These formats were compatible with the Firefox browser used in the pilot. For future projects, choosing formats that are compatible with the targeted browsers will be necessary.