

Tracker Business Intelligence Toolkit User Guide

John Conlon, Verticon, Inc.

Tracker Business Intelligence Toolkit User Guide

by John Conlon

Published November 1, 2012

Copyright © 2010, 2011, 2012 Verticon, Inc.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Tracker Business Intelligence Toolkit is free software: you can redistribute it and/or modify it under the terms of the EPL Eclipse Public License as published by the Eclipse Foundation, version 1.0.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Eclipse Foundation Software User Agreement for more details.

A copy of the Eclipse Foundation Software User Agreement is included in the Section entitled "Eclipse Foundation Software User Agreement" and an online version of the EPL Eclipse Public License version 1.0 can be found at <http://www.eclipse.org/org/documents/epl-v10.php>

Table of Contents

| | |
|-----------------------------------------------------|----|
| The Need for Livestock Information Management | x |
| 1. Overview | 2 |
| What is the Tracker Toolkit? | 2 |
| Tracker Business Intelligence Features | 2 |
| Tracker Business Intelligence Solutions | 2 |
| Project Goals | 2 |
| 2. Use Cases | 3 |
| Tracker Solution Use-Cases | 3 |
| Data Collection | 3 |
| Documentation | 4 |
| Management | 5 |
| Analysis and Reporting | 6 |
| Collaboration | 7 |
| 3. Tracker Product Suite | 9 |
| 4. Tracker Desktop BI Toolkit | 10 |
| Overview | 10 |
| What is the Tracker Desktop BI Toolkit? | 10 |
| Getting Started | 10 |
| Workbench Instances, Workspaces, and Projects | 12 |
| Perspective, Editors and Views | 12 |
| 5. Provisioned Editors | 15 |
| 6. Premises Editor | 16 |
| Overview | 16 |
| What is the Premises Editor? | 16 |
| Project Goals | 16 |
| Functional Overview | 16 |
| Premises Documents | 16 |
| Creating Premises Documents | 17 |
| Editing Premises Documents | 20 |
| Two Types of Editor | 20 |
| Complementary Views | 22 |
| Entering Date and Time | 23 |
| Master Detail Views | 24 |
| Animals View | 26 |
| Events View | 27 |
| Services | 28 |
| Location Services | 29 |
| TrackerStore Services | 32 |
| Bulk Event Processing | 32 |
| Bulk Loading New Events and Animals | 32 |
| Animal Document Templates | 33 |
| Tag IdNumber Files | 33 |
| Usage and Operations | 34 |
| Selecting and Targeting Elements | 35 |
| Premises Action Commands | 37 |
| Premises Data Model | 47 |
| Data Model Core Elements | 47 |
| Data Model Animal Elements | 48 |
| Data Model Event Elements | 49 |
| Custom Events | 51 |
| Creating Custom Events | 51 |

| | |
|--------------------------------------------------------------|----|
| Using Custom Events in Premises Documents | 53 |
| Premises Editor Use Cases | 57 |
| 7. Metatype Model Editor | 63 |
| Overview | 63 |
| What is the Metatype Model Editor? | 63 |
| Project Goals | 63 |
| Functional Overview | 63 |
| Metatype Data Model and Model Editor Use Cases | 64 |
| Metatype Documents | 64 |
| Metatype Data Model | 65 |
| Data Model Core Elements | 65 |
| Editing Metatype Documents | 67 |
| Creating a new Document | 67 |
| 8. Geography Editor | 68 |
| Overview | 68 |
| What is the Geography Editor? | 68 |
| Project Goals | 68 |
| Functional Overview | 68 |
| Geography Editor Use Cases | 68 |
| Geography Documents | 69 |
| Geography Data Model | 69 |
| Editing Geography Documents | 70 |
| Creating a new Document | 70 |
| 9. Agriculture Editor and Agriculture Location Service | 72 |
| Overview | 72 |
| What is the Agriculture Editor and Location Service? | 72 |
| Project Goals | 72 |
| Functional Overview | 73 |
| Agriculture Documents | 73 |
| Editing Agriculture Documents | 74 |
| Creating a new Document | 74 |
| Loading Additional Resources into the Document | 75 |
| Completing an Agriculture Document | 77 |
| Agriculture Data Model | 78 |
| Data Model Core Elements | 79 |
| Agriculture Editor Use Cases | 79 |
| Location Service Use Cases | 80 |
| Configuring the Location Service | 81 |
| Agriculture Documents as Input to the Location Service | 81 |
| Configuring a Project with an Agriculture Nature | 81 |
| Using the Location Service | 82 |
| 10. Fair Editor | 84 |
| Overview | 84 |
| What is the Fair Editor? | 84 |
| Project Goals | 84 |
| Functional Overview | 85 |
| Fair Documents | 86 |
| Editing Fair Documents | 86 |
| Creating a new Fair Document | 86 |
| Loading Additional Resources into the Document | 87 |
| Opening a Fair Document | 89 |
| Fair Action Commands | 90 |
| Fair Data Model | 94 |
| Fair Data Model Core Elements | 94 |

| | |
|-----------------------------------------------------------------------|-----|
| Fair Editor Use Cases | 96 |
| Prefair Use Cases | 97 |
| WeighIn Use Cases | 98 |
| Check-In Use Cases | 99 |
| 11. Provisioned Features | 101 |
| 12. WLIC Export | 102 |
| Overview | 102 |
| What is WLIC Exporting? | 102 |
| How to Export WLIC Information from Tracker Premises Documents? | 102 |
| 13. Fair XLS Import and Export | 103 |
| Fair XLS Import and Export Overview | 103 |
| Exporting Functionality | 103 |
| Exporting Information as Microsoft Excel documents | 103 |
| 14. OCL Query | 104 |
| Overview | 104 |
| OCL Interpreter | 104 |
| Queries View | 107 |
| 15. Tag ID Reader | 110 |
| Tag ID Reader Overview | 110 |
| Adding Event Transactions to Documents | 110 |
| File Reader | 110 |
| Usage and operations | 110 |
| 16. iRouter Premises Gateway | 112 |
| Premises EID Reader Gateway | 113 |
| RFID/EID Reader Background | 113 |
| EID Number Events to iRouter EID Products | 113 |
| Premises Measurement Event Gateway | 113 |
| iRouter Measurement and Position Products to EventAdmin Events | 114 |
| Gateway EventHandler | 114 |
| Animal History Events | 114 |
| EventAdmin Events to Animal History Events | 114 |
| Animal Life Data Gateway | 115 |
| Premises Gateway Examples | 115 |
| Recording Animal Location Example | 115 |
| Recording Animal Weight Example | 116 |
| 17. iRouter Premises Common Producers | 119 |
| Premises Event Producer | 120 |
| Premises Animal Producer | 121 |
| 18. MongoDB Tracker Store and Location Service | 122 |
| Overview | 122 |
| Provisioned Features | 122 |
| Server Installation and Administration | 123 |
| Administration Tasks | 123 |
| Configuring MongoDB iRouter Services | 128 |
| iRouter Consumer Service | 134 |
| iRouter Example | 136 |
| A. GNU Free Documentation License | 139 |
| B. Eclipse Foundation Software User Agreement | 145 |
| Eclipse Foundation Software User Agreement | 145 |

List of Figures

| | |
|--------------------------------------------------------------------------|----|
| 2.1. Data Collection Use-Case Diagram | 4 |
| 2.2. Documentation Use-Case Diagram | 5 |
| 2.3. Management Use-Case Diagram | 6 |
| 2.4. Analysis and Reporting Use-Case Diagram | 7 |
| 2.5. Collaboration Use-Case Diagram | 8 |
| 4.1. Tracker BI Toolkit Product Features | 11 |
| 4.2. Tracker Perspective | 13 |
| 6.1. Create a new Project | 17 |
| 6.2. Create a new Premises Document | 18 |
| 6.3. A new Premises Document | 19 |
| 6.4. Premises Form Editor | 20 |
| 6.5. Choosing a Premises Editor type to Open a Premises Document | 21 |
| 6.6. Premises Editor Synchronized with Views | 22 |
| 6.7. Date and Time Picker Dialog | 23 |
| 6.8. Animals View Horizontal Layout | 24 |
| 6.9. Animals View Vertical Layout | 25 |
| 6.10. Animals View Filtered Master Elements | 26 |
| 6.11. Animals View Filtered Detail Attributes | 26 |
| 6.12. Animals View | 27 |
| 6.13. Events View | 28 |
| 6.14. Animals View before Geolocation Resolution | 30 |
| 6.15. Animals View after Geolocation Resolution | 31 |
| 6.16. Selecting Animal Events | 36 |
| 6.17. After Invoking Target Selection | 37 |
| 6.18. Invoking Resource Actions from Project Explorer | 38 |
| 6.19. Invoking Element Actions from Premises Editor Selection Page | 41 |
| 6.20. Invoking Element Actions from Animals View Master Table | 42 |
| 6.21. Premises Data Model Core Elements | 47 |
| 6.22. Premises Data Model Animal Elements | 49 |
| 6.23. Premises Data Model Event Elements (part 1) | 50 |
| 6.24. Premises Data Model Event Elements (part 2) | 51 |
| 6.25. Defining Custom Events | 53 |
| 6.26. Load Resource Dialog | 54 |
| 6.27. Premises Document Loaded with a Metatype Resource | 55 |
| 6.28. Adding an Event in the Premises Editor | 56 |
| 6.29. Adding and Event in the Event View | 56 |
| 6.30. Modifying Attributes in the Events View | 57 |
| 6.31. Premises Editor Overview | 58 |
| 6.32. Setting Up the Premises Editor | 59 |
| 6.33. Updating Documents | 60 |
| 6.34. Navigating in the Editor | 61 |
| 6.35. Working with Connected Readers | 62 |
| 7.1. Metatype Data Model and Metatype Editor Use Cases | 64 |
| 7.2. Metatype Model Document | 66 |
| 7.3. New Wizard for Creating a New Metatype Document | 67 |
| 8.1. Geography Editor Use Cases | 69 |
| 8.2. New Wizard for Creating a New Geography Document | 70 |
| 8.3. New Geography Document Wizard Prompt | 71 |
| 9.1. New Wizard for Creating a New Agriculture Document | 74 |
| 9.2. New Agriculture Document Wizard ()Model Object Prompt | 75 |
| 9.3. Load Resource Dialog | 76 |

| | |
|--------------------------------------------------------------------------------|-----|
| 9.4. Agriculture Project with Loaded Resources | 77 |
| 9.5. Picklist of Premises | 78 |
| 9.6. Agriculture Data Model Elements | 79 |
| 9.7. Agriculture Editor Overview | 80 |
| 9.8. Location Service Overview | 81 |
| 9.9. Configuring Project with an Agriculture Nature | 82 |
| 9.10. Example of Resolving an Animal Location from Coordinates | 83 |
| 10.1. New Fair Document Wizard Model Object Prompt | 87 |
| 10.2. Load Resource Dialog | 88 |
| 10.3. Fair Project with Loaded Resources | 89 |
| 10.4. Opening a Fair Transaction Editor | 90 |
| 10.5. Invoking Element Actions from Fair Editor | 91 |
| 10.6. Invoking Element Actions from People View Master Table | 92 |
| 10.7. Fair Core Data Model Elements | 95 |
| 10.8. Hierarchy of Exhibits | 96 |
| 10.9. Fair Editor Overview | 97 |
| 10.10. Preparing for the fair | 98 |
| 10.11. Animal Weigh-In | 99 |
| 10.12. Animal Check-In | 100 |
| 14.1. Invoking OCL Console | 105 |
| 14.2. OCL Query Console | 106 |
| 14.3. Queries View | 108 |
| 16.1. GPS Gateway Example | 116 |
| 16.2. Animal Weight Gateway Example | 118 |
| 17.1. Premises Event Producer and Premises Animal Producer Example | 120 |
| 18.1. Admin Data Model Elements | 124 |
| 18.2. Loading an Admin document from Admin Editor Selection Page | 126 |
| 18.3. Register a Premises from the Premises Editor Selection Page | 128 |
| 18.4. Edit an iRouter Configuration | 129 |
| 18.5. iRouter Editor Factories | 130 |
| 18.6. Creating a New Mongo TrackerStore Service | 131 |
| 18.7. Configuring a TrackerStore Service | 132 |
| 18.8. TrackerStore Service Running and Shown in the iRouter Monitor View | 134 |
| 18.9. TrackerStore Service iRouter Network Product Flows | 136 |

List of Tables

| | |
|-----------------------------------------------------------------------------|-----|
| 6.1. Format Patterns for Text Entry | 24 |
| 6.2. File Resource Action Commands | 39 |
| 6.3. Folder Resource Action Commands | 40 |
| 6.4. Action Commands on all Elements | 43 |
| 6.5. Action Commands on Animal Elements | 44 |
| 6.6. Action Commands on Event Elements | 45 |
| 6.7. Action Commands on Premises Elements | 46 |
| 6.8. Core Elements of the Premises Model | 48 |
| 7.1. Core Elements of the Metatype Model | 66 |
| 9.1. Core Elements of the Agriculture Model | 79 |
| 10.1. Action Commands on all Elements | 93 |
| 10.2. Action Commands on Person, Animals, and Lots | 94 |
| 10.3. Core Elements of the Premises Model | 95 |
| 13.1. Worksheets exported by the <i>Fair Registration XLS file...</i> | 103 |
| 13.2. Worksheets exported by <i>Fair Registration XLS file...</i> | 103 |
| 16.1. Premises Gateway Component Factories | 112 |
| 17.1. Premises Common Component Factories | 120 |
| 18.1. Elements of the Admin Model | 125 |
| 18.2. Admin Editor Document Element Action Commands | 127 |
| 18.3. Mongo Tracker Store iRouter Consumer | 136 |

List of Examples

| | |
|------------------------------------------------------------------------------|-----|
| 6.1. A Time Stamped Tag IdNumber File Name | 34 |
| 14.1. Listing of animals and the number of events associated with each. | 107 |
| 14.2. Listing of animals and the set of events associated with each. | 107 |
| 14.3. Listing of animalIds and the WeighIn events associated with each. | 107 |
| 14.4. Listing of animalIds and the Weights associated with each. | 107 |
| 14.5. Find the animal with active tag 985152002153189. | 109 |
| 14.6. Find all MilkTest Events with Somatic Cell Counts > 200 | 109 |
| 14.7. Find all Animals with undefined MovedIn Source Pin attributes | 109 |

The Need for Livestock Information Management

Either for hunting or husbandry, human beings have always observed and remembered animal movements. Today, with larger herds and more interaction between producers, infrastructure providers, and process auditors more animal movements mean more information for the producer, the provider, and the auditor to manage.

Markets reach peak efficiency when buyers and sellers have an equal *or symmetric* view of product information.¹ How can animal agriculture markets, which are today constrained by an *asymmetric information* exchange between buyers and sellers be tuned to meet peak potential?

Tracker is a suite of Business Intelligence² solutions for the

- Documentation;
- Management;
- Analysis;
- Reporting;
- and Market Transaction Exchange;

of Animal Livestock *Information*.

¹In 2001, the Nobel Prize in Economics was awarded to George Akerlof, Michael Spence, and Joseph E. Stiglitz "for their analysis of markets constrained by asymmetric information."

²the process of gathering information about a business or industry matter; a broad range of applications and technologies for gathering, storing, analyzing, and providing access to data to help make business decisions; abbr. BI

Colophon

Production versions of this documentation were produced with the *Docbook* [<http://www.docbook.org/>] *DSSSL* style sheets, [http://en.wikipedia.org/wiki/Document_Style_Semantics_and_Specification_Language] called by the *Apache Ant build tool* [<http://ant.apache.org/>], from within an *Eclipse IDE* [<http://www.eclipse.org/>].

Chapter 1. Overview

What is the Tracker Toolkit?

Tracker is a Business Intelligence Toolkit (BI) and suite of Business Intelligence featured solutions for animal agriculture professionals. The Tracker BI Toolkit is an Eclipse based *network aware rich client platform* that can be provisioned with one or more BI Features.

Tracker Business Intelligence Features

Tracker Business Intelligence Features provide historical, current, and predictive views of animal agriculture business operations. Tracker BI Features are used by *Animal Agriculture producers, infrastructure providers* and *process control auditors* to

- Document;
- Manage;
- Analyze;
- Report;
- and Inter-exchange business information.

Tracker Business Intelligence Solutions

Tracker BI Features are cost effective solutions for;

- recording animal event histories;
- maintaining production efficiency;
- improving herd quality;
- documenting process controls;
- recording animal movements;
- and sharing animal history information.

Project Goals

The Tracker project was undertaken to provide;

- a. a common platform container for a suite of Business Intelligence solutions useful for Livestock producers and infrastructure providers;
- b. free Open Source technologies to encourage widespread vendor integration and user deployment of generic Livestock Business Intelligence solutions;
- c. lower costing custom solutions for Livestock producers, infrastructure providers and process auditors;
- d. create ubiquitous livestock event history inter-exchange solutions via associations and marketplaces.

Chapter 2. Use Cases

Tracker Solution Use-Cases

This section utilizes *use-case diagrams* to describe Tracker functionality from the users¹ point of view. Tracker user functionality can be organized into the following six categories:

1. Data Collection;
2. Documentation;
3. Management;
4. Analysis;
5. Reporting;
6. and Collaboration.

Note

To accommodate the largest number of users, in the most open way possible, Tracker applications support multiple Operating Systems. Tracker application runtime binaries are available for the following leading Operating System platforms:

- Microsoft Windows™;
- Linux;
- and Apple MacOS™.

Data Collection

Offering a value added suite of information solutions to livestock professionals initially requires the collection of a diverse data set prior to processing and reporting.

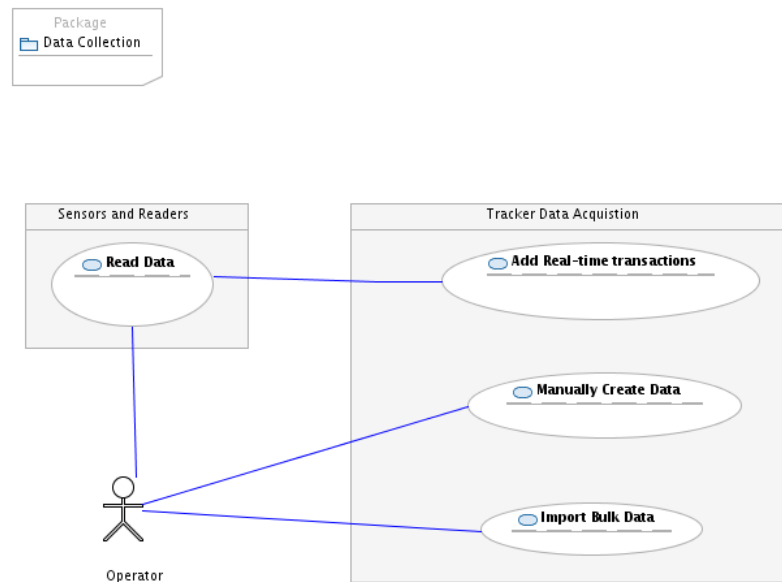
Tracker provides three ways to collect livestock data:

- manual data entry by the Tracker operator;
- importation of bulk data from external files;
- and input of real-time electronic transactions from sensors and readers.

Data Collection functionality is shown in the Figure 2.1, “Data Collection Use-Case Diagram”².

¹Users of Tracker are typically Livestock *Producers* operating the Tracker Desktop applications, but they may also be auditors, infrastructure providers or Livestock Information Analysts. The section will use the terminology *Operator* and *Information Technology Analyst* to distinguish between field operators and information analysts.

² More detailed use-case diagrams are documented in the latter chapters on Provisioned Editors and Features.

Figure 2.1. Data Collection Use-Case Diagram

Documentation

After data is collected, the collected data itself needs to be structured, consolidated and persisted. Next users need to confirm that the results of this process is a *valid model* of the real world process they are trying to capture.

Tracker applications consolidate, structure and persist data to light weight industry standard [<http://en.wikipedia.org/wiki/XMI>] document files. Since files can be attached and sent via email or other file sharing technologies using standard based document files for data storage provides Tracker users significant benefits over storing data in heavy weight relational databases or proprietary document formats. If business requirements dictate the need for multi-user data storage or the storage of large volumes of premises data, the optional Tracker Store feature offers an easy to deploy, robust and highly scalable data storage solution.³

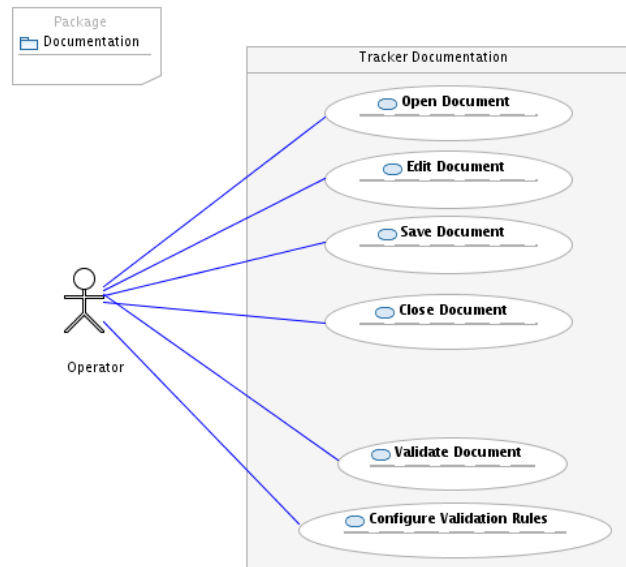
Structuring and consolidating data to documents is the first step of the process, but users need tools to confirm that documents are *valid* models of real world processes. For the document to be considered valid for a real world process, rules for judging validity must be controlled by the user.⁴

Documentation functionality is shown in the Figure 2.2, “Documentation Use-Case Diagram”

³ Industry standard text based data documents offer greater mobility than database data storage and greater processing, tooling, analysis, reporting and collaborating options than proprietary document formats. Open standard document formats are the key to many of the subsequent use-case scenarios presented in this section.

This documentation uses the term *Tracker Data Documents* as a shorthand for the more technically precise term *EMF Ecore* [<http://www.eclipse.org/modeling/emf/>] *Model Instance* document.

⁴ Document validation basically insures that data contained in documents meet user requirements. Since user requirements may change based on a specific project or livestock scenario, the user must be able to modify and change these rules.

Figure 2.2. Documentation Use-Case Diagram

Management

Tracker data documents are created and used within a specific livestock business context. Within this context Tracker documents are often associated with other files and folders of files.⁵

To address the requirement for grouping files into work contexts Tracker supports the concept of a *Project*.⁶

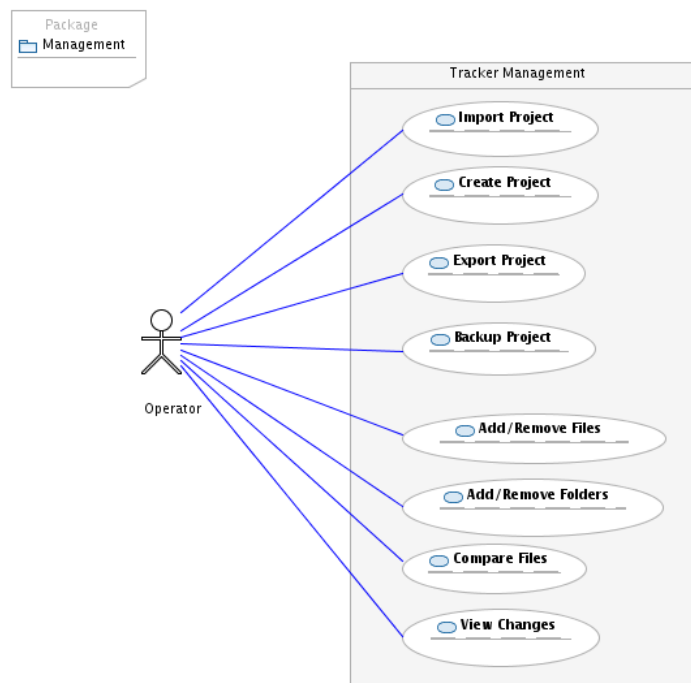
A Tracker project is a conceptual container for managing *related* files, folders, documents, reports and images. Files in Tracker projects can be created, organized in folders, removed, deleted and edited. Files in Tracker Projects can also be compared to each other and the editing histories of files can also be viewed and changes compared. Projects (and all the contained files) can be archived, exported, and imported into the Tracker.⁷

Management functionality is shown in the Figure 2.3, “Management Use-Case Diagram”

⁵ For example: Tracker data documents may be associated with the raw data files that were initially imported into the Tracker documents; Tracker data documents are often accompanied by report design files used for running reports; and there may even be multiple Tracker document files associated with different data viewpoints of a particular work activity.

⁶ Eclipse also supports a higher or parent level of container above that of a project container. This higher level container is called an Eclipse *workspace* and it exists to hold user settings and contain multiple projects. A Tracker user can work with Multiple Workspaces each containing multiple Projects. For the typical Tracker user, they can elect to only use the default workspace and afterwards ignore the concept of workspaces.

⁷ Tracker also supports simple project management tools for managing ToDo tasks and network backup of projects. Tools for more sophisticated project management and file revision control are available as Eclipse features that can also be added to the Tracker.

Figure 2.3. Management Use-Case Diagram

Analysis and Reporting

To create information useful to end users, Tracker data documents can be used as data sources to Eclipse Business Intelligence Reporting Tool [<http://www.eclipse.org/birt/phenix/intro/>] (BIRT) reports.⁸

BIRT reports are created by Livestock Information Technology Analysts.⁹ After analyzing specific livestock information problems these analysts utilize the BIRT Report Designer tool to specify how to extract and create information from Tracker data documents. These specifications result in a set of BIRT report design documents and report libraries. Report design documents and libraries are distributed to Tracker End Users who use them to run reports against one or more data documents in order to visualize and export information.¹⁰

In addition to enabling Tracker users to run reports against Tracker data documents, Information Technology Analysts can further enable Tracker End Users by crafting report based solutions that integrate data from Tracker data documents and any other source of BIRT data. Besides Tracker data documents BIRT reports can access and visualize data from:

1. SQL Relational Databases through any JDBC driver;
2. Eclipse Modeling Framework EMF Ecore data models;
3. Non-standard data models through custom code;
4. Flat file data using standard delimiters;
5. XML files or streams using XPath syntax to specify columns.

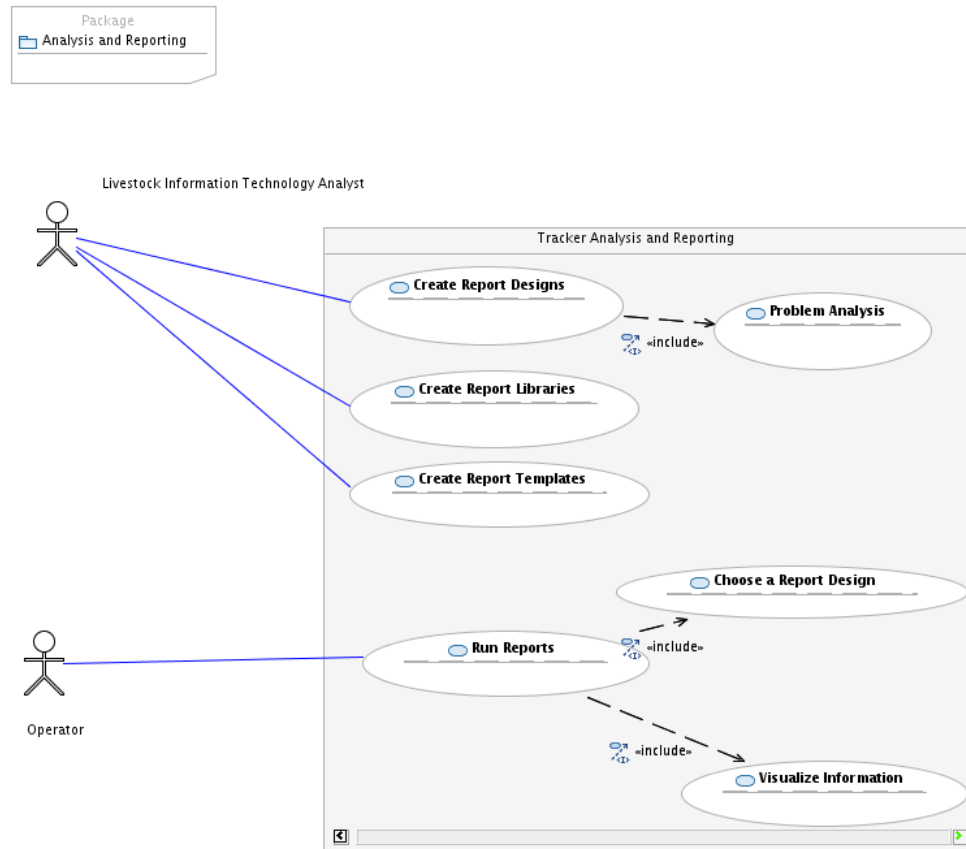
⁸TrackerStore can also be used as a data source for BIRT reports.

⁹Also called Business Analysts or Report Designers.

¹⁰Tracker end users would be able to run reports but would not have access to the Report Designer tool.

Analysis and Reporting functionality is shown in the Figure 2.4, “Analysis and Reporting Use-Case Diagram”

Figure 2.4. Analysis and Reporting Use-Case Diagram



Collaboration

Because Tracker data documents are small industry standard data containers they are ideal for exchanging and sharing *validated* livestock information. Tracker documents are easily shared between individual users, groups of users, business collaborators and central auditing authorities.¹¹

Collaboration technology options range from manual file over media transfers,¹² to common Internet file transfer utilities,¹³ to a mediated collaboration service that offers publish and request services to Tracker clients.¹⁴

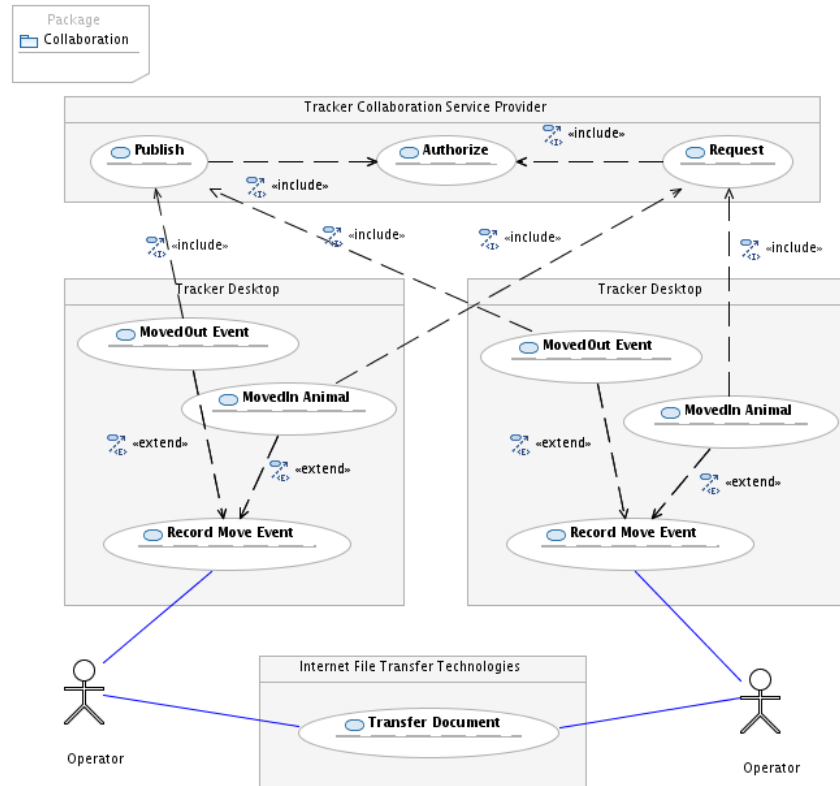
Collaboration functionality is shown in the Figure 2.5, “Collaboration Use-Case Diagram”

¹¹BIRT reports can also easily be created and shared with other users and collaborators.

¹² For example documents or document fragments can be exchanged on usb flash drives or flash memory cards.

¹³Email, FTP, and Web Server uploading and downloading.

¹⁴ The TrackerStore is an example of such a service. One or more TrackerStore Collaboration Service Providers could be utilized by Tracker clients. Example administrators of these service providers could be industry associations or government auditing authorities.

Figure 2.5. Collaboration Use-Case Diagram

Chapter 3. Tracker Product Suite

The Tracker Product Suite is a comprehensive suite of livestock information products. The Tracker product suite consists of the following products:

- Tracker BI Toolkit Desktop;
- Tracker Information Exchange Cloud Service;
- Tracker Web Client;
- Tracker Mobile Client;
- Tracker iRouter Appliance.

Chapter 4. Tracker Desktop BI Toolkit

Overview

What is the Tracker Desktop BI Toolkit?

The Tracker Desktop BI Toolkit is a desktop based Business Intelligence (BI) solutions toolkit for animal agriculture professionals. The Tracker Desktop Toolkit is a *network aware rich client platform* that can be installed with one or more user selected BI features.

Tracker Desktop Toolkit is based on the Eclipse Rich Client Platform (RCP), the Eclipse Modeling Framework (EMF) and Open System Gateway Initiative (OSGi) technologies.

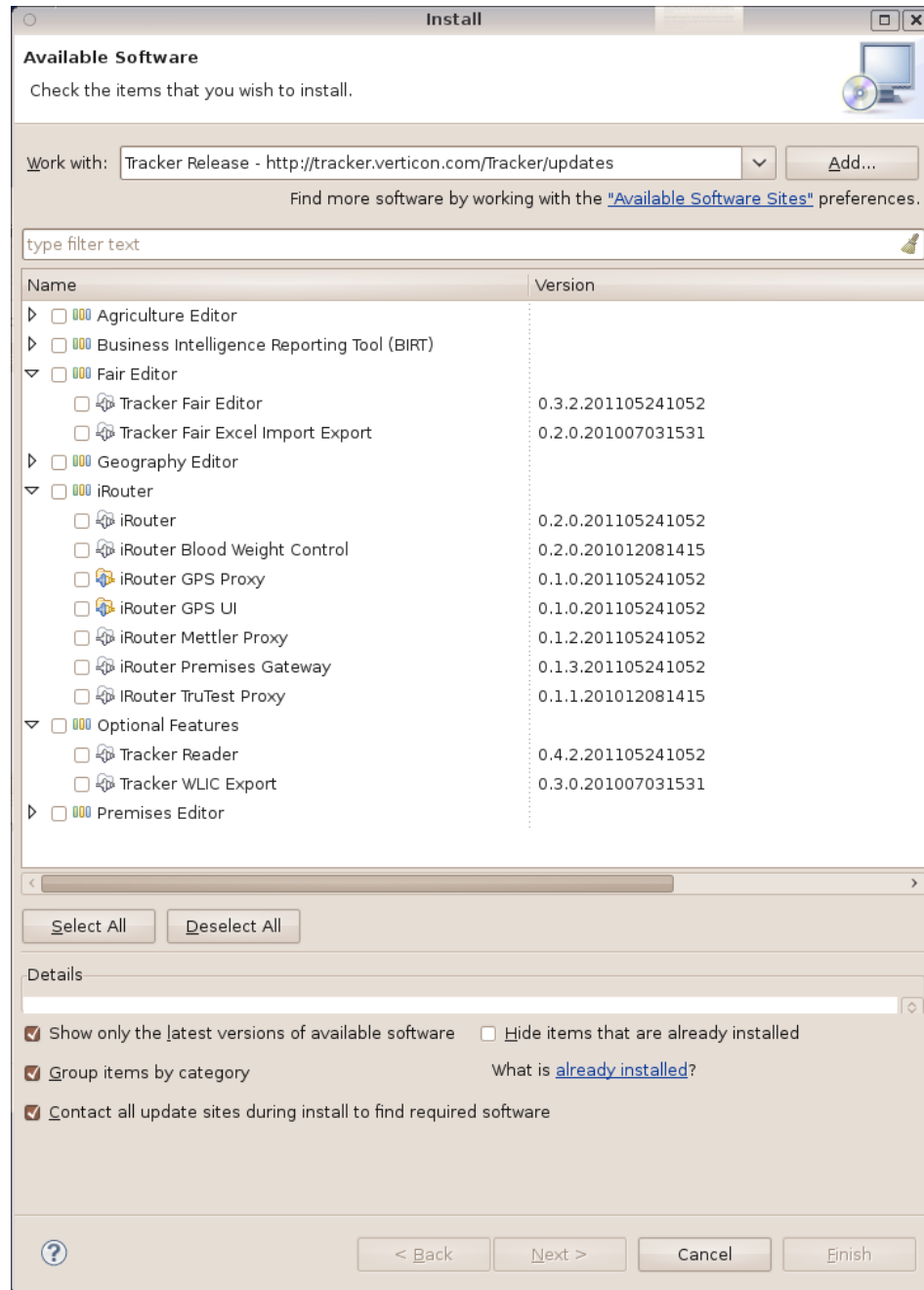
Getting Started

Go to the Tracker website *Tracker Website* [<http://tracker.verticon.com/downloads/index.html>] and follow the installation instructions.

Once the Tracker BI Toolkit product is installed users can easily add tools to meet their unique information management requirements. Tracker BI Toolkit can be extended with one or more optional features:

- features that contain editors for creating and managing business specific information;
- features that contain task specific utilities and tools for complimenting information;
- features that contain business intelligence reporting tools for presenting and analyzing information;

To add tools, go to the *Help* menu and select *Install New Software*, From the Install Wizard, in the *Work with* selection box, select the entry *Tracker Release* - <http://tracker.verticon.com/Tracker/updates>.

Figure 4.1. Tracker BI Toolkit Product Features**Note**

You can also use a cheat sheet to assist with adding features. Go to the *Help* menu and selecting Cheat Sheets. From the *Tracker* entry select *Adding Tracker Features to the Tracker Desktop Toolkit*.

Workbench Instances, Workspaces, and Projects

The main focus of activity in the Tracker Application is the editing of Animal Premises Documents. In order to discuss the mechanics of editing Tracker Animal Premises Documents, some definitions are in order:

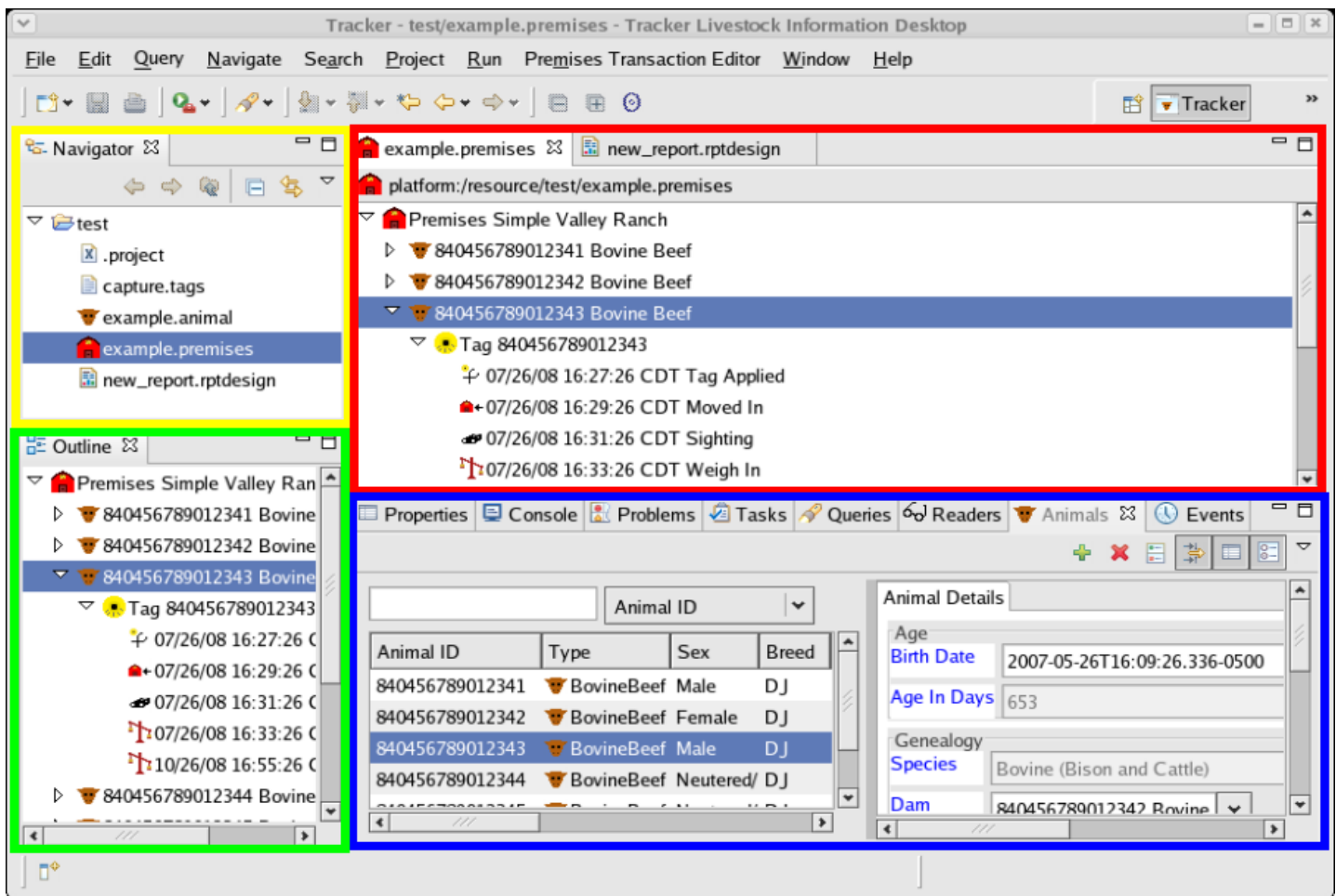
- A Tracker Desktop Toolkit *installation* is a location on your disk that contains a distribution of the Tracker. When you unpack a Tracker distribution to a location on your disk, you effectively create a local platform installation at that location. The installation contains a set of tools that were packaged together by the organization that created the distribution. Each of the tools in the platform installation contributes some functionality to the Tracker. You may have several platform installations on your machine, each of which may contain a different set of tools.
- A Tracker *workbench instance* is a runtime instance of the Tracker. Every time you launch the Tracker, a workbench instance is created, making the tools in the platform installation from which the instance was loaded available for use.

For more information, see the Workbook User Guide concept definition for *Workbench* [<http://help.eclipse.org/ganymede/index.jsp?topic=/org.eclipse.platform.doc.user/concepts/concepts-2.htm>].

- A Tracker *workspace* is a location on your disk that is associated with a running workbench instance. It contains the resources (Tracker Documents, files, folders, etc.) that comprise the *projects* you work on. The tools contained in the platform installation from which the workbench instance was launched can be used to work with the projects in the workspace. Each workbench instance has exclusive access to its associated workspace: concurrently running workbench instances can not share the same workspace.

Perspective, Editors and Views

Within the Tracker Workbench, Documents are edited, manipulated, and examined with data model specific Document Editors and Views. To assist you as you work with Documents, a Tracker *Perspective* visualizes and positions these Tracker specific Editors and Views within the Workbench as shown in Figure 4.2, “Tracker Perspective”.

Figure 4.2. Tracker Perspective

For more information, see the Workbook User Guide concept definitions for *Editor* [<http://help.eclipse.org/ganymede/index.jsp?topic=/org.eclipse.platform.doc.user/concepts/concepts-6.htm>], *View* [<http://help.eclipse.org/ganymede/index.jsp?topic=/org.eclipse.platform.doc.user/concepts/concepts-5.htm>], and *Perspective* [<http://help.eclipse.org/ganymede/index.jsp?topic=/org.eclipse.platform.doc.user/concepts/concepts-4.htm>].

The Tracker Perspective organizes the workbench into four areas, outlined in colors in the above figure.

- The resource navigation area, highlighted in yellow, provides is the standard Resource Navigator view which presents the tree of Projects, Folders, Files and other resources in the workspace.
- An area for visualizing editors, highlighted in red, where Tracker Document Editors and other Resource Editors display.

Optional Editors

Displayed is the optional Premises Editor that may be installed (called *provisioned*) by the user.

- The *Outline View* [<http://help.eclipse.org/ganymede/index.jsp?topic=/org.eclipse.platform.doc.user/reference/ref-30.htm>] , highlighted in green, displays an outline of a structured file that is currently open in the editor area, and lists structural elements.

- d. The support views area, highlighted in blue, displays the Properties, Console, Problems, Tasks, Queries, Readers, Animals, and Events views.

Optional Views

The Queries, Readers, Animals, and Events views are optional views that may be provisioned by the user.

The *Properties View* [<http://help.eclipse.org/ganymede/index.jsp?topic=/org.eclipse.platform.doc.user/reference/ref-29.htm>] displays read-only information, such as the properties of a file resource and information that can be edited, such as the properties of an Entry in a Document.

The Console View displays program output information, such as error or warning messages.

The Problems View displays problems related to workbench or resources.

The Tasks View displays user reminders of things to do.

The Ocl Queries View displays a list of quick selection mechanisms for selecting and find information in Tracker Documents.

The Readers View displays a list of configured Tag ID Readers.

Chapter 5. Provisioned Editors

Livestock producers, processors and auditors require tools for modeling, recording, editing, and reporting on livestock operational information. These tools must be able to meet requirements common to each type of livestock operation but also be capable of adapting to specific use cases. For each use case the essential consideration is how to structure or *model* the operational data to ensure vital information can be efficiently captured, stored, analyzed, and utilized in an operationally valuable way.

Common operational use cases are analyzed, then modeled by Tracker business analysts resulting in concise *data model* specifications. From these data model specifications Tracker engineers craft use case specific editors, tools, and utilities which are then delivered as Tracker features to end users.

Tracker Business Intelligence Features consist of the following data model specific editors:

- Premises Editor, is the primary Tracker Livestock editor and its associated data model is often extended by the other data model editors;
- The Geography Editor is an open source desktop solution for authoring, managing and integrating state of the art Geographic Information System GIS technologies based on the *OpenGIS* [<http://www.opengeospatial.org/>] Key Hole Markup Language KML;
- Fair Editor, extends the Premises Editor to model and document people and animals at a Livestock Exhibition. The Premises Editor and the Fair Editor are the initial Tracker editors.
- Metatype Model Editor, is a used to create configuration documents that extend the functionality of core Tracker Features;
- Agriculture Editor is a solution for documenting, managing and visualizing geographic livestock herd distributions, individual animal locations, and animal movements within and between premises locations.

The Agriculture Editor and Location Service is a suite of tools that integrates state of the art Geographic Information System GIS technologies based on the *OpenGIS* [<http://www.opengeospatial.org/>] Key Hole Markup Language KML with Tracker Premises livestock management information;

- Custom Editors, are editors used to create customer specific data model documents that meet unique customer use-case requirements.

Chapter 6. Premises Editor

Overview

What is the Premises Editor?

The Premises Editor is a comprehensive open desktop solution for documenting and managing livestock premises, animal life data, and animal event history information.

Project Goals

The Premises Editor project was undertaken to provide;

- a. A simple to use editor for creating premises, animal and event history documentation for livestock producers and infrastructure providers;
- b. A low cost, yet extensible documentation solution to meet the needs of both large and small scale livestock operations;
- c. An industry standard document format for livestock premises information, animal life data and animal event history inter-exchange.

Functional Overview

The Premises Editor is a tool to create and edit Tracker Premises document. Premises documents specify Premises details, animal life data and animal event histories.

Premises details are manually added to premises documents by the user, but Animal life data and Event histories can be added to Tracker Premises Documents;

- a. As individual animals by the user of the Premise Editor, using either the form based Premises Views or with the generic Properties Viewer;
- b. In bulk, by importing lists of animals, through Tracker's Bulk Event Processing Utilities;
- c. In real time, from tag ID readers, through the Tracker Readers Feature.

See the following section Editing Premises Documents for details on Tracker Readers Feature and Premises document editing.

Note

The Premises Editor comes with a set of predefined event types that are suitable for documenting most livestock histories. For recording unique premises specific events, the Premises Editor also provides a facility to define custom event types. Once defined, these custom event types can be used along with the predefined event types to record animal histories.

Premises Documents

Whether it is word processing documents, spreadsheet documents, or presentation documents, today's computer users are most familiar with software applications that create and manipulate documents that describe specific instances of information.

The Premises Editor enables users to create, transform and edit Livestock Premises documents that represent Animal Premises. The attributes of the premises, the animals in the premises and the events associated with these animals are specified in Premises documents.

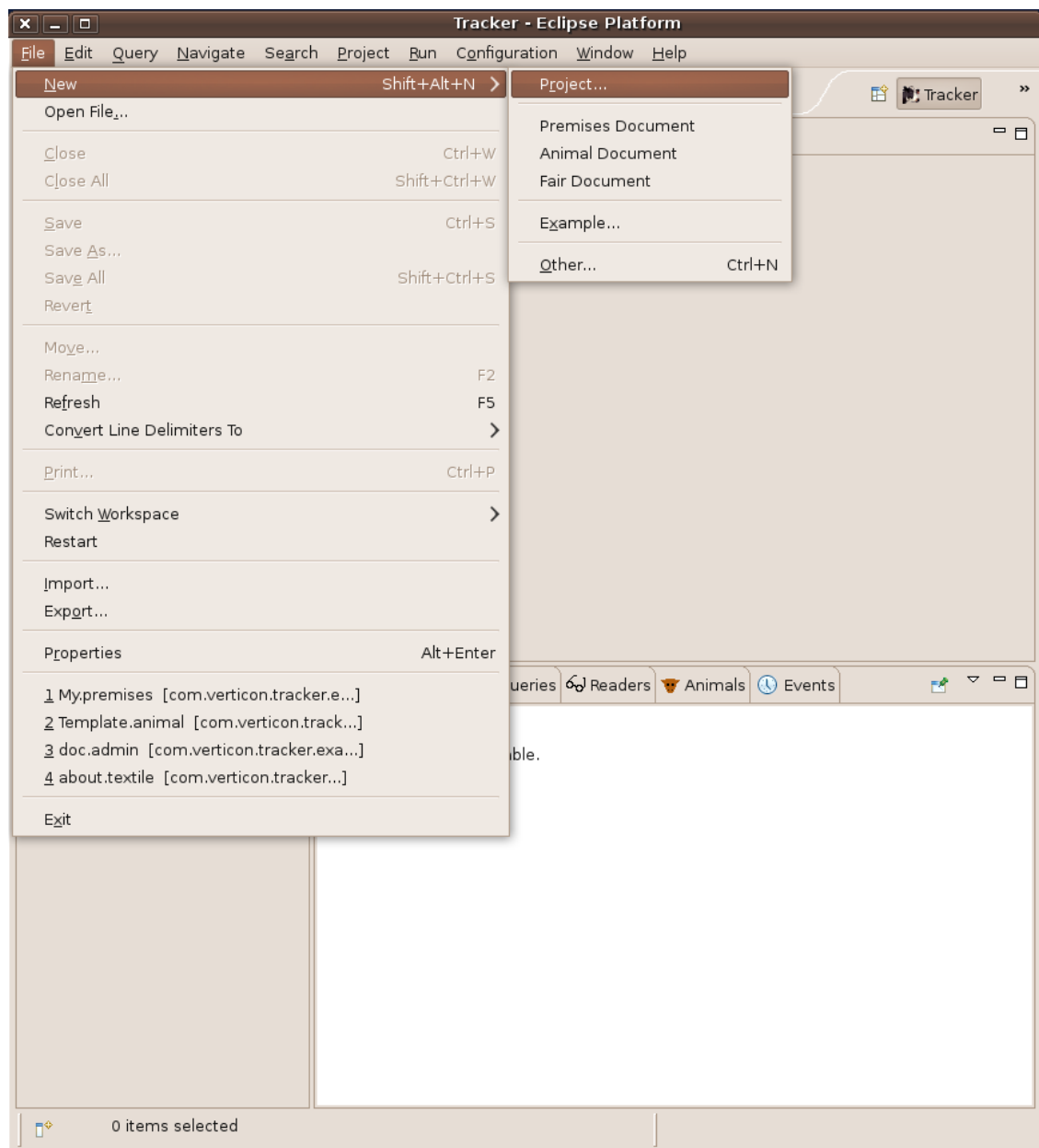
Premises document files utilize a .premises file extension to distinguish them from other types of files.

Creating Premises Documents

This section will walk you through a few simple steps for the creation of an example Premises document.

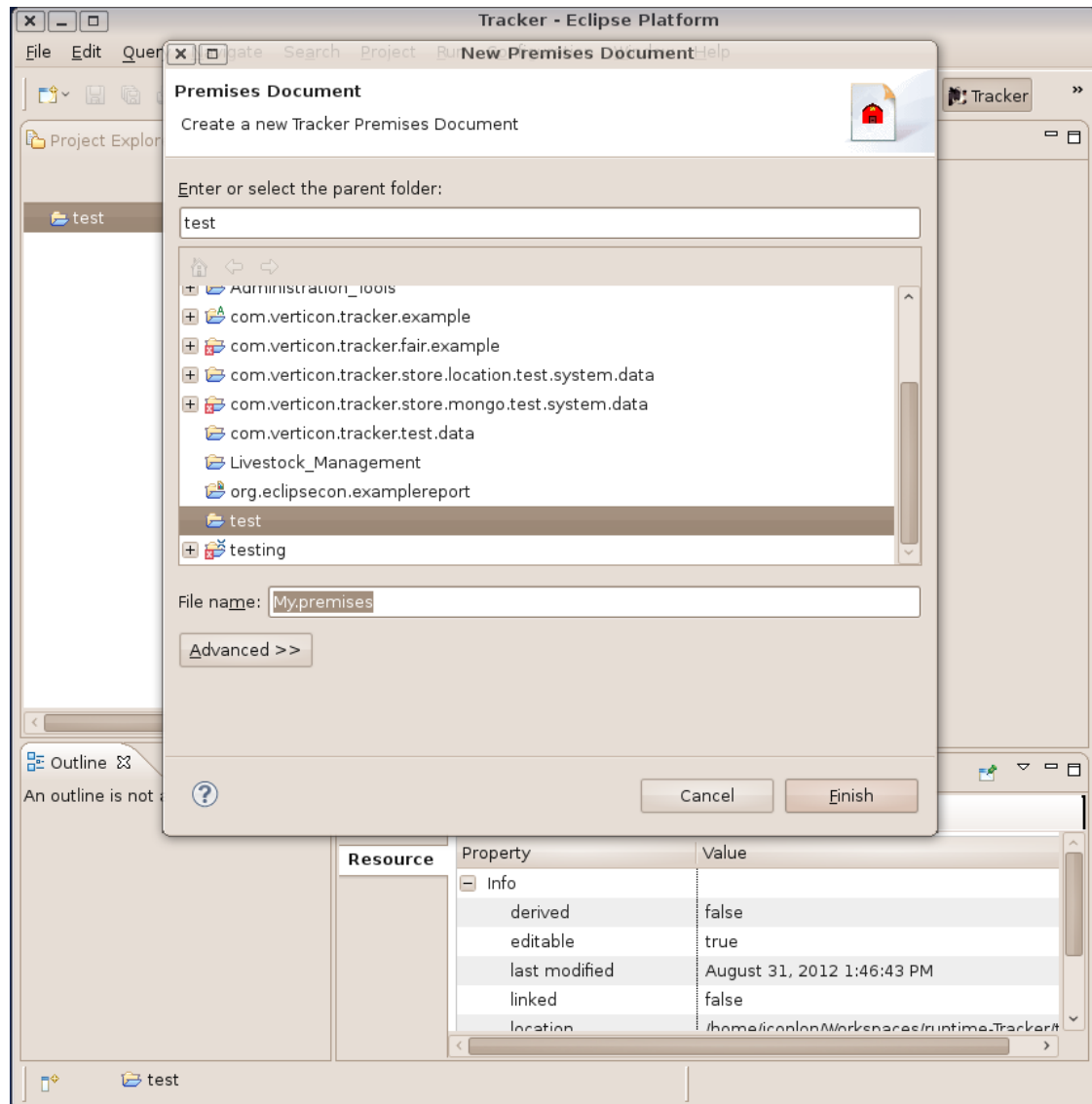
If you haven't done so, create a project first. From the 'File' menu select 'New' and 'Project' and give it the name 'test'.

Figure 6.1. Create a new Project

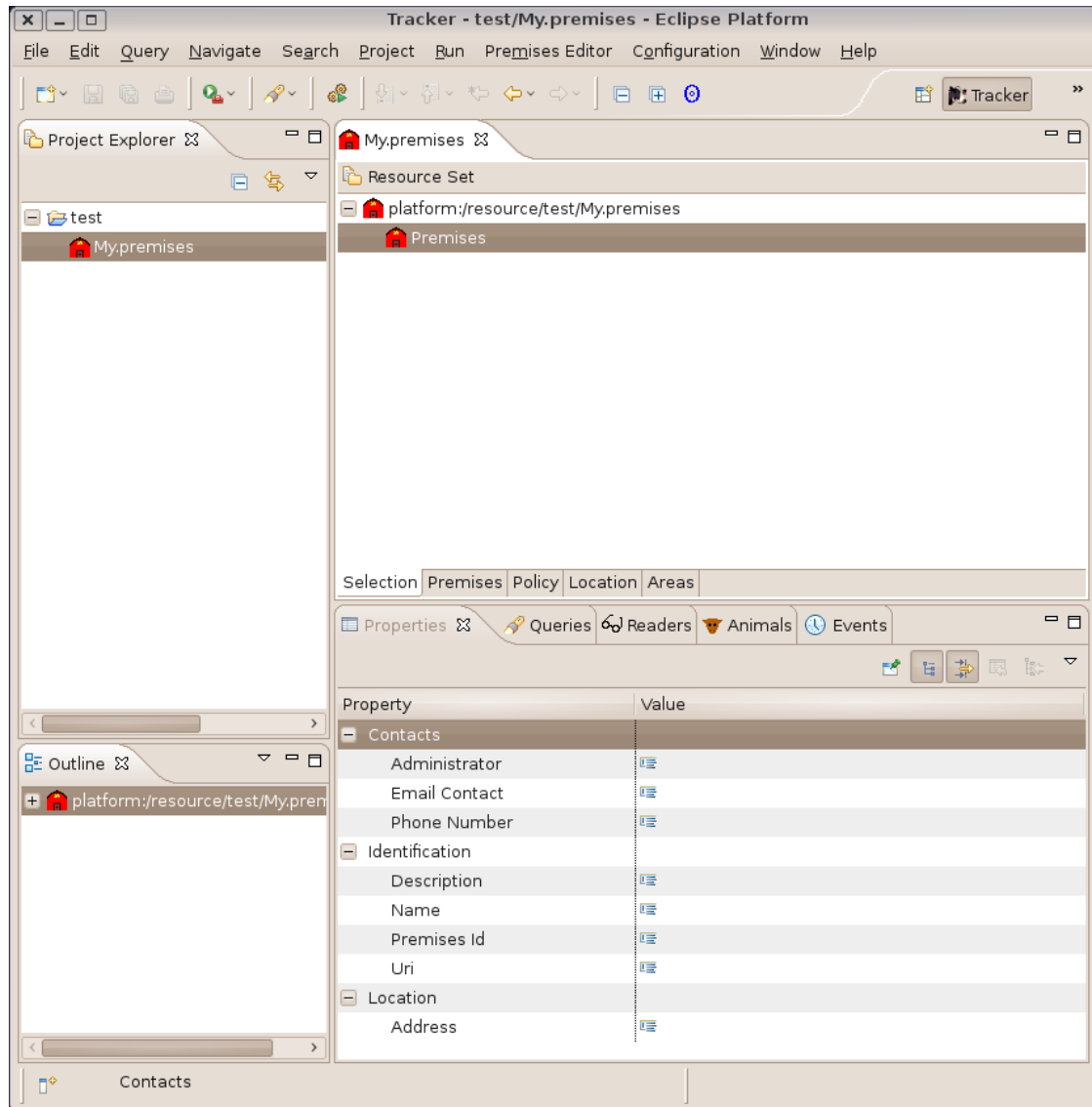


Creation of new project artifacts like files and documents follow the same pattern as the creation of the project. From the 'File' menu select 'New' then select 'Premises Document'. Take the default name for the document and place it under the project folder named 'test'. Figure 6.2, “Create a new Premises Document” shows the screens for creating a Premises document.

Figure 6.2. Create a new Premises Document



Your workspace should now look something like Figure 6.3, “A new Premises Document”. If you don't see the Properties View in lower part of the desktop, select the Premises element, right click, and from the popup menu select the last entry 'Show Properties View'.

Figure 6.3. A new Premises Document

The 'Properties View' is a generic editor for all document elements¹. You can use it to add, delete or replace element attributes. To edit an elements properties in the 'Properties View', select it first in an editors 'Selection' window.

In addition to the main 'Selection' editor, the Premises editor offers three form editors to edit Premises, Policy, Location, and Area elements and their attributes. Figure 6.4, "Premises Form Editor" shows the 'Premises' form editor in contrast with the generic 'Properties View'².

¹ Any Tracker document element once selected, can be edited with the 'Properties View'.

² Either the form editors or the 'Properties View' may be used for editing.

Figure 6.4. Premises Form Editor

Tracker - test/My.premises - Eclipse Platform

File Edit Query Navigate Search Project Run Premises Editor Configuration Window Help

Pr *My.premises

test

test

platform/resources

Selection Premises Policy Location Areas

Properties Queries Readers Animals Events

| Property | Value |
|----------------|---------------------|
| Contacts | |
| Administrator | Jack Condor |
| Email Contact | |
| Phone Number | |
| Identification | |
| Description | An example premises |
| Name | Hornby Hollow Farm |
| Premises Id | |
| Location | |
| Address | |

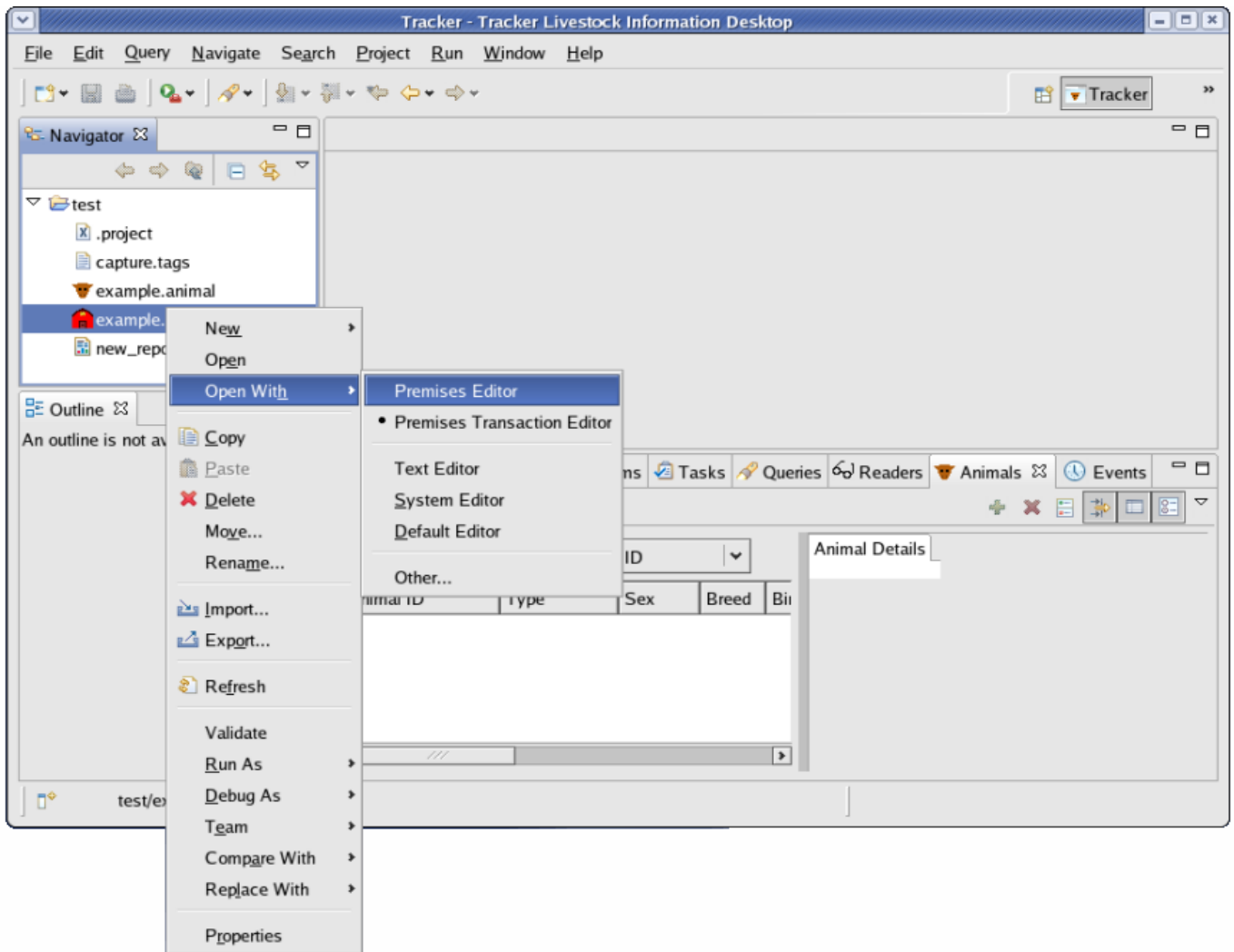
Selected Object: Pr... Hornby Hollow Farm

Editing Premises Documents

Two Types of Editor

Two types of editors are used to manipulate Premises and Animal Documents³, Premises Editor and Premises Transaction Editor as shown in Figure 6.5, “Choosing a Premises Editor type to Open a Premises Document”.

³Premises and Animal documents created or edited in one editor type can later be opened and edited with the other.

Figure 6.5. Choosing a Premises Editor type to Open a Premises Document

- a. The Premises Editor is the default editor. It is the editor that is opened when a document is created. The Premises Editor is used for creating, manipulating, navigating and interpreting Premises and Animal Documents when there is no requirement for interfacing with external devices.⁴
- b. The Premises Transaction Editor can be used in a similar way to the Premises Editor, but it has additional functionality to support the interfacing with external devices like RFID tag readers.

Reader Feature

The Readers Feature is an optional Feature that may be installed (called *provisioned*) by the user. The Readers Feature allows the Tracker BI Toolkit User to control and configure connections to one or more Tag ID Event Readers enabling them to independently add Event Transactions to Premises Documents.

⁴The Premises Editor also provides user friendly forms based editors for editing Premises, Policy, Location, and Area elements and attributes. These form based editors are not supported in the Premises Transaction Editor. So with the Premises Transaction Editor the Properties View will have to be used to edit these elements and their attributes.

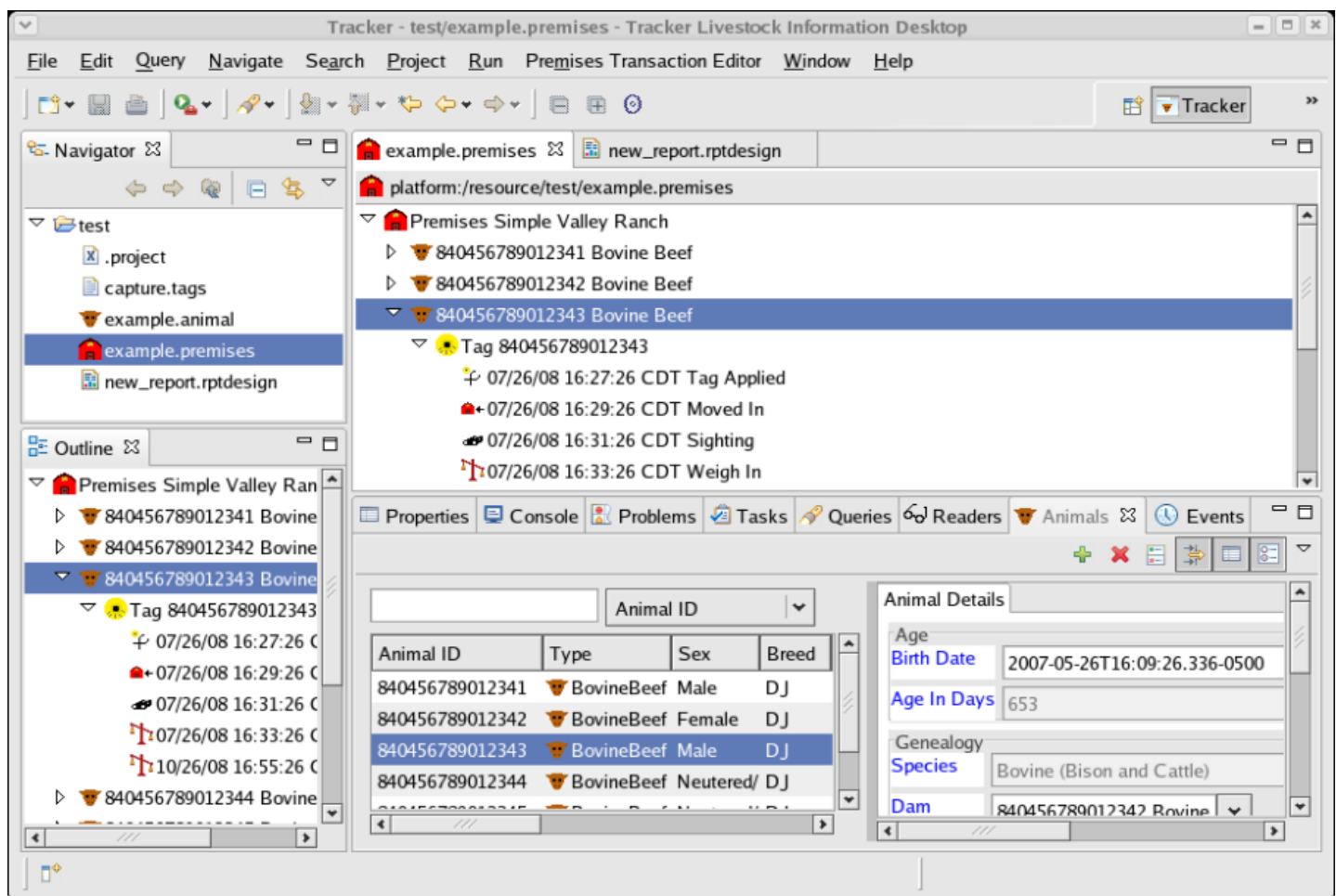
Note

Both the Premises Editor and the Premises Transaction Editor can be used for bulk event processing. For more information on bulk even processing see the section called “Bulk Event Processing”.

Complementary Views

The Premises Editor is complemented by several Eclipse and Premises Editor Feature Specific views, each giving the user a unique way to *view and edit* the information in a Premises Document. Selecting information in the Premises Editor will synchronize selections in complementary views as shown in Figure 6.6, “Premises Editor Synchronized with Views”.

Figure 6.6. Premises Editor Synchronized with Views



The *Outline View* [<http://help.eclipse.org/ganymede/index.jsp?topic=/org.eclipse.platform.doc.user/reference/ref-30.htm>] is synchronized with selections in the Premises Editor and the *Properties View* [<http://help.eclipse.org/ganymede/index.jsp?topic=/org.eclipse.platform.doc.user/reference/ref-29.htm>]. By selecting an element in the Outline View, that Element is also selected in Premises Editor and it's attributes are visualized in the Properties View.

Premises Editor selections are also synchronized with the Premises specific Animals view and Events views. These two views provide a *master-detail view of information in Premises documents*.

For more information on master details views see the section called “Master Detail Views”.

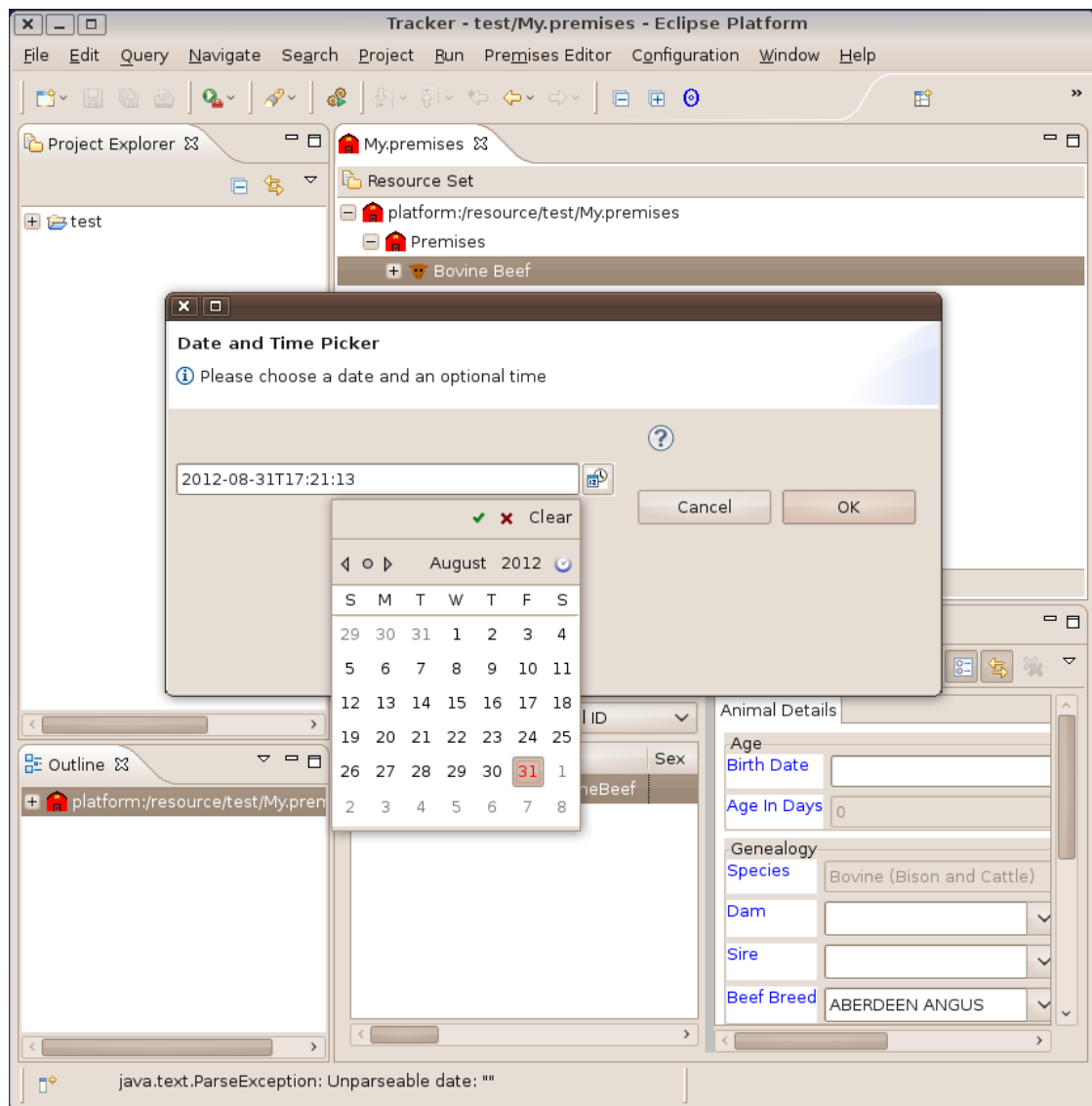
Note

The Tracker Model Editor also supports an advanced element selection and query functionality, for details see *Tracker OCL Query Feature*.

Entering Date and Time

When using the forms based editors to change date times, clicking on the text entry field of an attribute that is a DateTime should bring up a 'Date Time Picker' dialog. Figure 6.7, “Date and Time Picker Dialog” shows the Date Picker Dialog that was visualized after the user clicked on the Animal Birth Date text field in the details pane⁵ of the Animals View.

Figure 6.7. Date and Time Picker Dialog



⁵The next section discusses Master Details Views.

When manually entering Dates⁶ and DateTime attribute values in Premises documents the following formatting patterns are determinative:

Table 6.1. Format Patterns for Text Entry

| Format Pattern | Example |
|-----------------------------|------------------------------|
| yyyy-MM-dd'T'HH:mm:ss'.SSSZ | 2007-11-12T00:00:00.000-0600 |
| yyy-MM-dd'T'HH:mm:ss'.SSS | 2007-11-12T00:00:00.000 |
| yyyy-MM-dd'T'HH:mm:ss | 2007-11-12T00:00:00 |
| yyyy-MM-dd'T'HH:mm | 2007-11-12T00:00 |

Master Detail Views

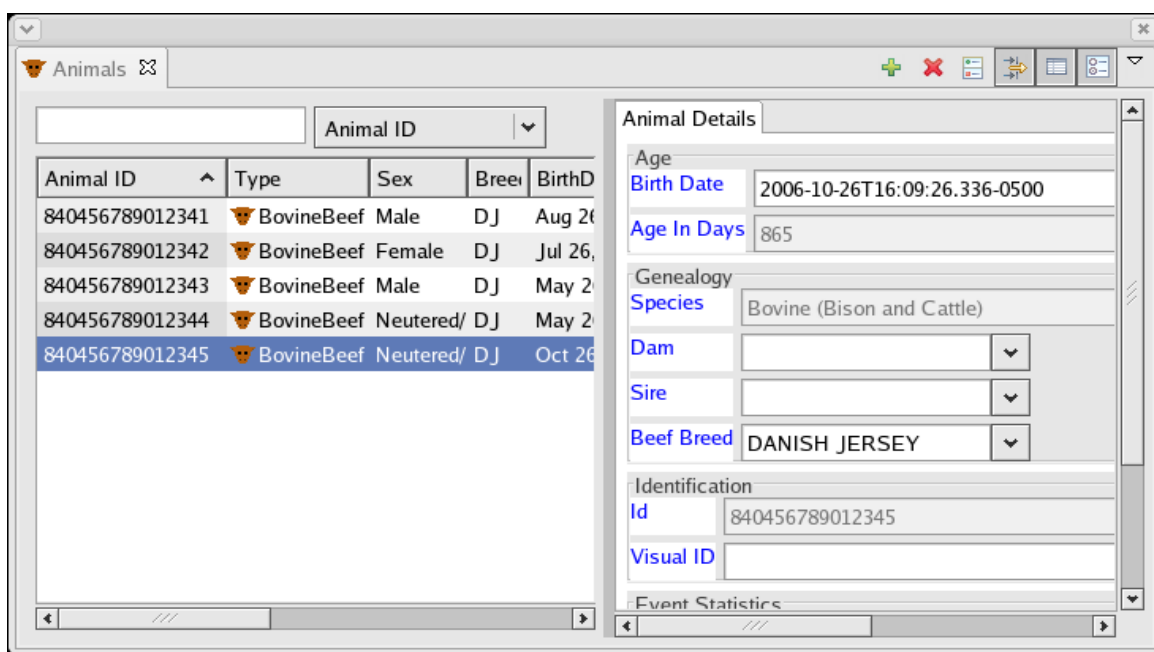
A master detail view is one where a master element and its related detail attributes are shown on the same view.

The master element is depicted as a row in a table of other master elements derived from the document. Columns can be rearranged by dragging the column headers to the desired location and master elements in the table can be sorted by clicking on the column header to use as a sort key. Clicking a second time will sort in the opposite direction.

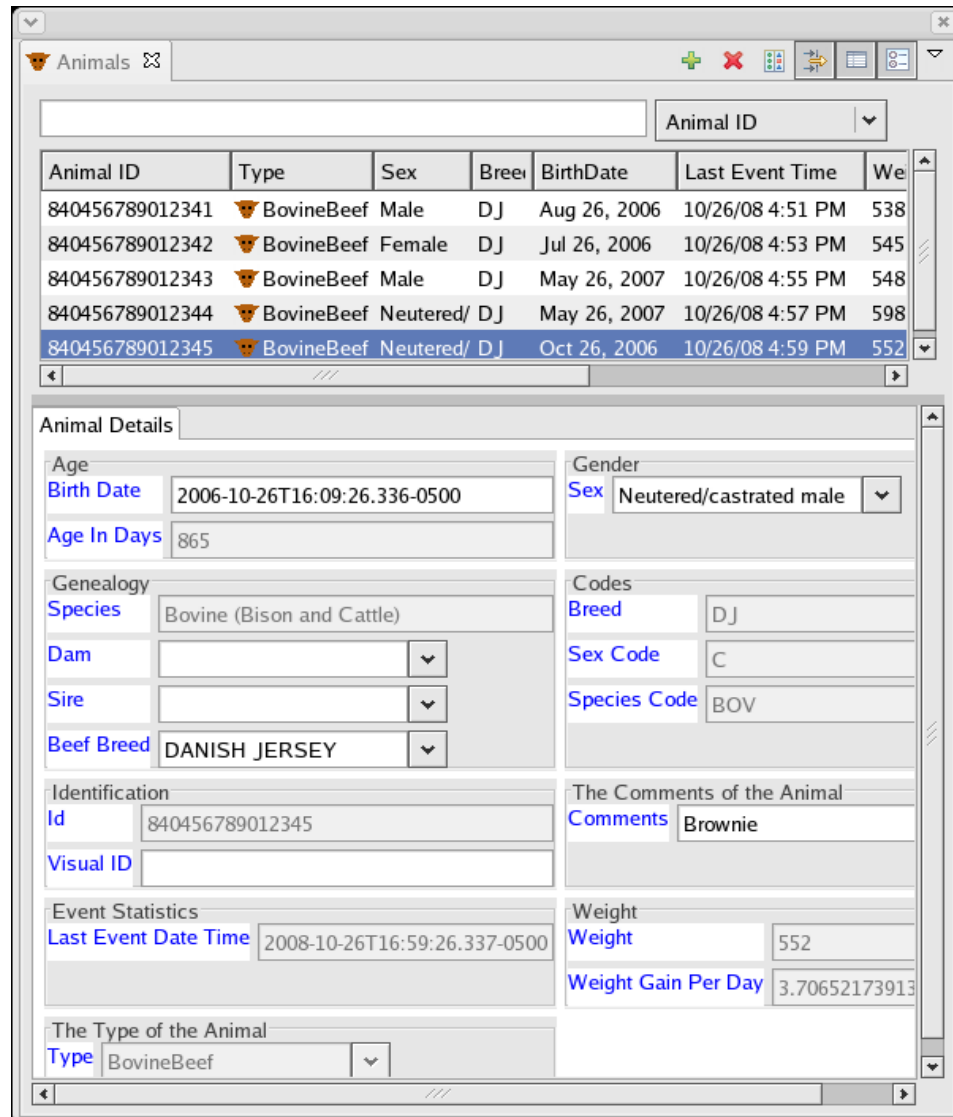
The Detail attributes are depicted within a form. The detail attributes in form may consist of read-write and read-only attributes. The detail attributes are displayed based on the current element row selection in the master table.

A master detail view can be displayed with a horizontal layout as shown in Figure 6.8, “Animals View Horizontal Layout” or with a vertical layout as shown in Figure 6.9, “Animals View Vertical Layout”.

Figure 6.8. Animals View Horizontal Layout



⁶The Animal BirthDate value is the only Date value that can be set to null by entering the text 0-0-0 in the Properties view or the Animals view.

Figure 6.9. Animals View Vertical Layout


The screenshot displays the 'Animals' view in the Premises Editor. At the top, there is a search bar and a dropdown for 'Animal ID'. Below this is a table listing animals with columns: Animal ID, Type, Sex, Breed, BirthDate, Last Event Time, and Weight. The table contains five rows, with the last row (ID 840456789012345) selected. Below the table is a detailed form for the selected animal, divided into several sections: 'Animal Details' (Age, Birth Date, Age In Days, Gender, Sex), 'Genealogy' (Species, Dam, Sire, Beef Breed), 'Identification' (Id, Visual ID), 'Event Statistics' (Last Event Date Time), 'The Comments of the Animal' (Comments), 'Weight' (Weight, Weight Gain Per Day), and 'The Type of the Animal' (Type).

| Animal ID | Type | Sex | Breed | BirthDate | Last Event Time | Weight |
|-----------------|------------|-----------|-------|--------------|------------------|--------|
| 840456789012341 | BovineBeef | Male | DJ | Aug 26, 2006 | 10/26/08 4:51 PM | 538 |
| 840456789012342 | BovineBeef | Female | DJ | Jul 26, 2006 | 10/26/08 4:53 PM | 545 |
| 840456789012343 | BovineBeef | Male | DJ | May 26, 2007 | 10/26/08 4:55 PM | 548 |
| 840456789012344 | BovineBeef | Neutered/ | DJ | May 26, 2007 | 10/26/08 4:57 PM | 598 |
| 840456789012345 | BovineBeef | Neutered/ | DJ | Oct 26, 2006 | 10/26/08 4:59 PM | 552 |

Animal Details

Age: Birth Date: 2006-10-26T16:09:26.336-0500, Age In Days: 865, Gender: Sex: Neutered/castrated male

Genealogy

Species: Bovine (Bison and Cattle), Dam: , Sire: , Beef Breed: DANISH JERSEY

Identification

Id: 840456789012345, Visual ID:

Event Statistics

Last Event Date Time: 2008-10-26T16:59:26.337-0500

The Comments of the Animal

Comments: Brownie

Weight

Weight: 552, Weight Gain Per Day: 3.70652173913

The Type of the Animal

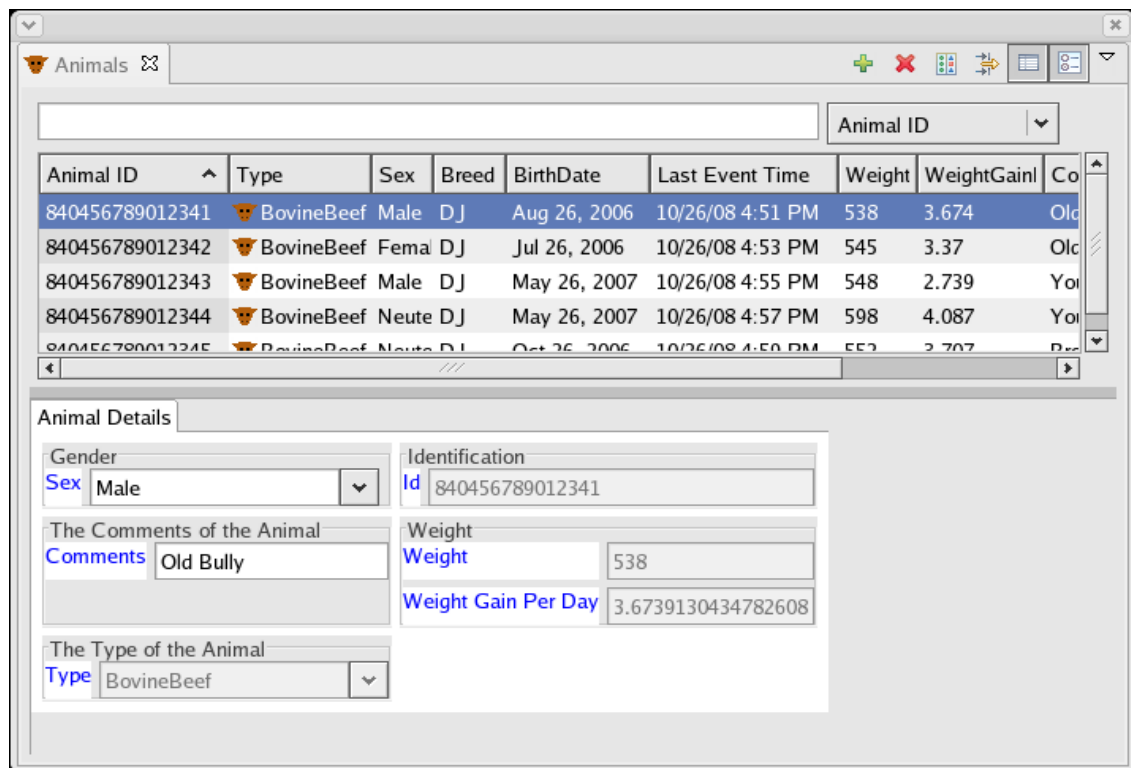
Type: BovineBeef

Elements displayed in the master table can be filtered by entering characters in the text field and the attribute selection control. Figure 6.10, “Animals View Filtered Master Elements ” shows the list of Animals filtered to display only Male animals.

Figure 6.10. Animals View Filtered Master Elements


| Animal ID | Type | Sex | Breed | BirthDate | Last Event Time | Weight | WeightGain | Comment |
|-----------------|------------|------|-------|--------------|------------------|--------|------------|-----------|
| 840456789012341 | BovineBeef | Male | DJ | Aug 26, 2006 | 10/26/08 4:51 PM | 538 | 3.674 | Old Bully |
| 840456789012343 | BovineBeef | Male | DJ | May 26, 2007 | 10/26/08 4:55 PM | 548 | 2.739 | Young Bil |

Using the fourth button in the view menu, attributes displayed in the detailed form can be filtered to show only basic attributes. Figure 6.11, “Animals View Filtered Detail Attributes” shows the effect of deselecting the 'Show Advanced Properties' button.

Figure 6.11. Animals View Filtered Detail Attributes


| Animal ID | Type | Sex | Breed | BirthDate | Last Event Time | Weight | WeightGain | Comment |
|-----------------|------------|--------|-------|--------------|------------------|--------|------------|-----------|
| 840456789012341 | BovineBeef | Male | DJ | Aug 26, 2006 | 10/26/08 4:51 PM | 538 | 3.674 | Old Bully |
| 840456789012342 | BovineBeef | Female | DJ | Jul 26, 2006 | 10/26/08 4:53 PM | 545 | 3.37 | Old Bully |
| 840456789012343 | BovineBeef | Male | DJ | May 26, 2007 | 10/26/08 4:55 PM | 548 | 2.739 | Young Bil |
| 840456789012344 | BovineBeef | Neute | DJ | May 26, 2007 | 10/26/08 4:57 PM | 598 | 4.087 | Young Bil |
| 840456789012345 | BovineBeef | Neute | DJ | Oct 26, 2006 | 10/26/08 4:50 PM | 552 | 2.707 | Old Bully |

Animal Details

Gender: Sex Male

Identification: Id 840456789012341

The Comments of the Animal: Comments Old Bully

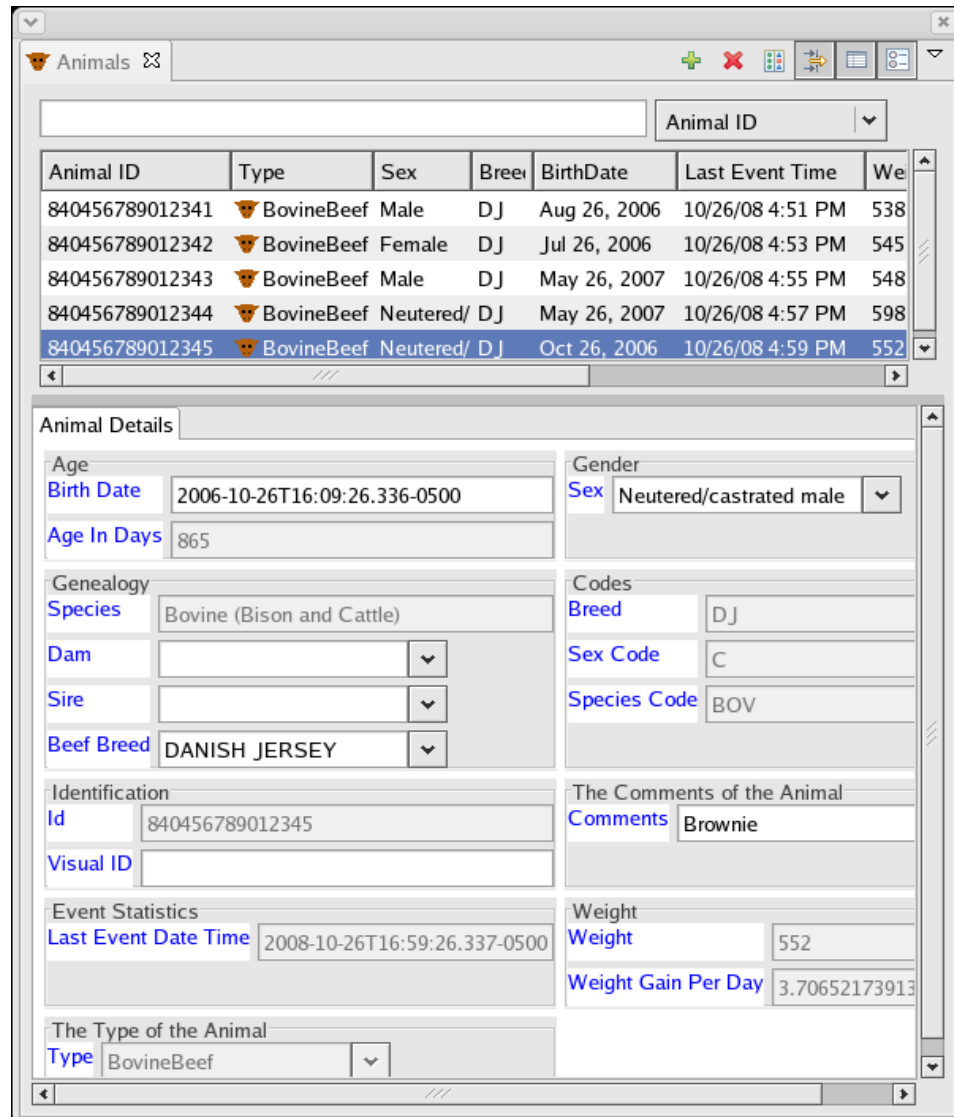
The Type of the Animal: Type BovineBeef

Weight: Weight 538

Weight Gain Per Day: Weight Gain Per Day 3.6739130434782608

Animals View

Figure 6.12, “Animals View ” is a master detail view of the Animals Elements in the Premise document being edited in the Premises Editor. Selections in the Premises Editor will synchronize downward to selections in the master table of the the Animals view. Selections in the master table of the Animals View will synchronize upward to the Premises Editor and the Outline View.

Figure 6.12. Animals View


The Animals View window displays a table of animals and a detailed view of the selected animal.

| Animal ID | Type | Sex | Breed | BirthDate | Last Event Time | Weight |
|-----------------|------------|-----------|-------|--------------|------------------|--------|
| 840456789012341 | BovineBeef | Male | DJ | Aug 26, 2006 | 10/26/08 4:51 PM | 538 |
| 840456789012342 | BovineBeef | Female | DJ | Jul 26, 2006 | 10/26/08 4:53 PM | 545 |
| 840456789012343 | BovineBeef | Male | DJ | May 26, 2007 | 10/26/08 4:55 PM | 548 |
| 840456789012344 | BovineBeef | Neutered/ | DJ | May 26, 2007 | 10/26/08 4:57 PM | 598 |
| 840456789012345 | BovineBeef | Neutered/ | DJ | Oct 26, 2006 | 10/26/08 4:59 PM | 552 |

Animal Details

Age: Birth Date: 2006-10-26T16:09:26.336-0500, Age In Days: 865

Gender: Sex: Neutered/castrated male

Genealogy: Species: Bovine (Bison and Cattle), Dam: , Sire: , Beef Breed: DANISH JERSEY

Codes: Breed: DJ, Sex Code: C, Species Code: BOV

Identification: Id: 840456789012345, Visual ID:

Event Statistics: Last Event Date Time: 2008-10-26T16:59:26.337-0500

The Comments of the Animal: Comments: Brownie

Weight: Weight: 552, Weight Gain Per Day: 3.70652173913

The Type of the Animal: Type: BovineBeef

Events View

Figure 6.13, “Events View ” is a master detail view of the Events Elements in the Premise document being edited in the Premises Editor. Selections in the Premises Editor will synchronize downward to selections in the master table of the the Events view. Selections in the master table of the Events View will synchronize upward to the Premises Editor and the Outline View.

Figure 6.13. Events View

| Date | Event | Animal | Tag ID | Comment |
|-----------------------|------------|-----------------------------|-----------------|---------|
| 10/26/08 16:59:26 CDT | WeighIn | Bovine Beef 840456789012345 | 840456789012345 | |
| 10/26/08 16:57:26 CDT | WeighIn | Bovine Beef 840456789012344 | 840456789012344 | |
| 10/26/08 16:55:26 CDT | WeighIn | Bovine Beef 840456789012343 | 840456789012343 | |
| 10/26/08 16:53:26 CDT | WeighIn | Bovine Beef 840456789012342 | 840456789012342 | |
| 10/26/08 16:51:26 CDT | WeighIn | Bovine Beef 840456789012341 | 840456789012341 | |
| 07/26/08 16:49:26 CDT | WeighIn | Bovine Beef 840456789012345 | 840456789012345 | |
| 07/26/08 16:47:26 CDT | Sighting | Bovine Beef 840456789012345 | 840456789012345 | |
| 07/26/08 16:45:26 CDT | MovedIn | Bovine Beef 840456789012345 | 840456789012345 | |
| 07/26/08 16:43:26 CDT | TagApplied | Bovine Beef 840456789012345 | 840456789012345 | |

| Event Details | |
|----------------------------------------------|----------------------------------------|
| Time of Event | Codes |
| Date Time 2008-10-26T16:59:26.337-0500 | Event Code 102 |
| Details | The Comments of the Event |
| Electronically Read <input type="checkbox"/> | Comments |
| Correction <input type="checkbox"/> | |
| Identification | Weight |
| Id 840456789012345 | Weight 552 |
| | Weight Gain Per Day 3.7065217391304346 |

Services

Tracker Premises Editor utilizes a Service Oriented Architecture SOA to provide the operator with set of dynamic services to help manage, analyze, store and retrieve Premises information.

Currently the Tracker Premises Editor provides the user with the following two types of dynamic services:

- Location Services - a suite of geolocation services for resolving locations;
- TrackerStore Services - a suite of storage services for storing and retrieving premises information.

Each type of service delivers a specific functionality, but that functionality may be provided by one or more dynamically provisioned and/or configured service implementations. For example there may be two types of Location services provisioned and configured that resolve a specific object location in one way and another object in an alternative way⁷.

⁷ When multiple location services are present the algorithm for resolving geolocations will request resolution from each location service implementation until the location is resolved.

Location Services

The Tracker Location Service is a global Tracker geolocation service for resolving the real-world geographic locations for Premises and Animals.

Together Premises Editor and the Tracker Location Service provide:

- Animal traceback for naming the foreign Premises where animals have resided before being moved into the current Premises⁸;
- Animal traceforward for naming the foreign Premises where animals have been moved to after they were movedOut of the Premises;
- Animal resident locator for identifying the location of animals within the current Premises;
- Dynamic listing of location names within the current Premises;

The Animal Location attribute is the last known location of the animal. It is derived from the last (most recent) geolocation event recorded for the Animal. The Premises Editor provides pulldown menus on an Animal or Premises⁹ for invoking the resolution of Geolocation names by the Location Service. The following names are resolved:

- Source Name on all MovedIn events;
- Destination Name on all MovedOut events;
- Location¹⁰ on Animal;
- Location Name on all Position events¹¹;

⁸Fundementally Location Services is a geolocation name resolution service. Animal traceback and traceforward requires an animal to contain a complete event history so that location identifiers can be resolved as names. To insure that animals moved into a premises retain a complete event history see the TrackerStore Services.

⁹See section on Premises Editor Actions for details.

¹⁰This is the current location of the animal.

¹¹Position Events with coordinates residing in the local premise have the Location attribute resolved directly by the Premises Editor.

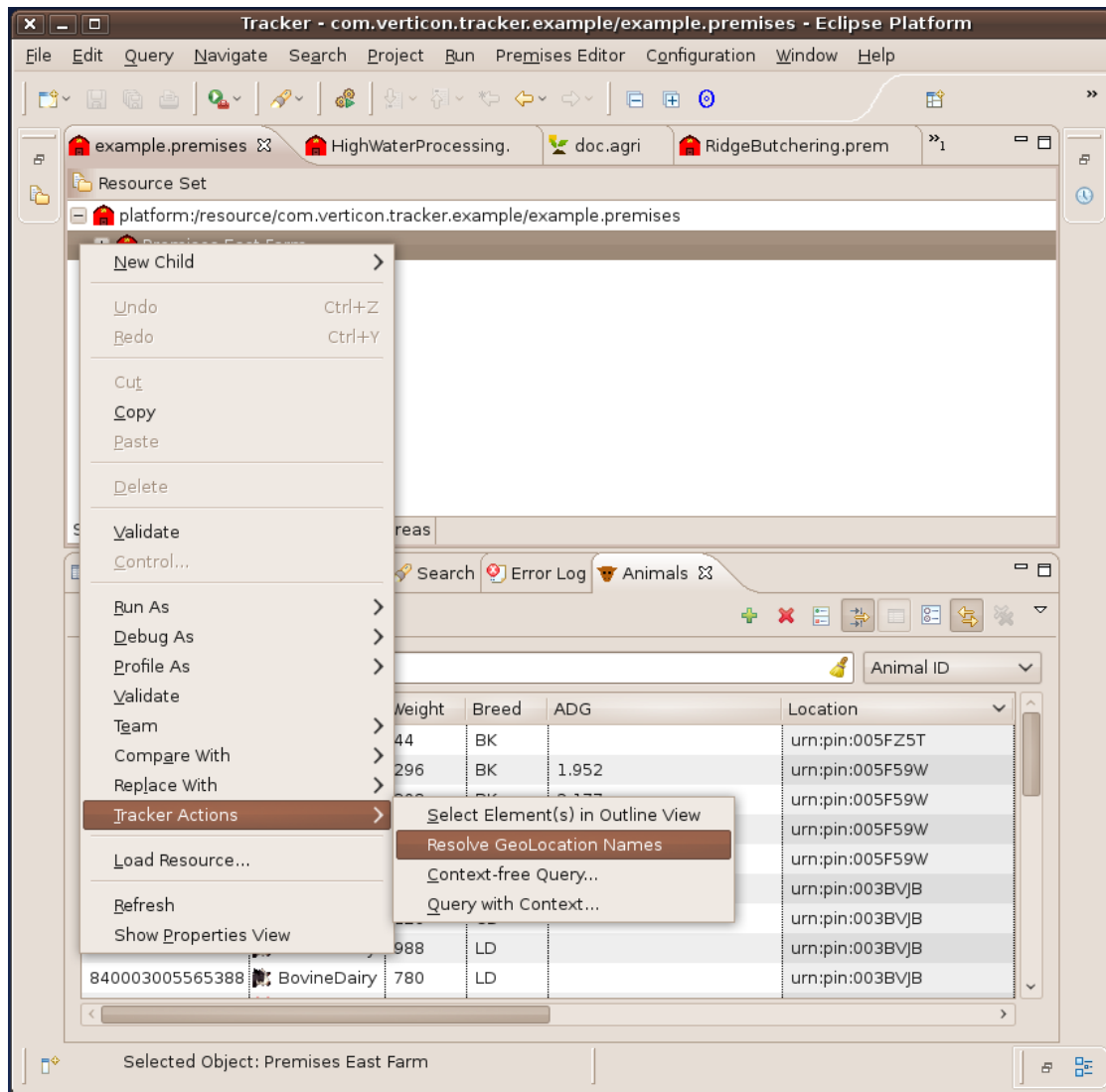
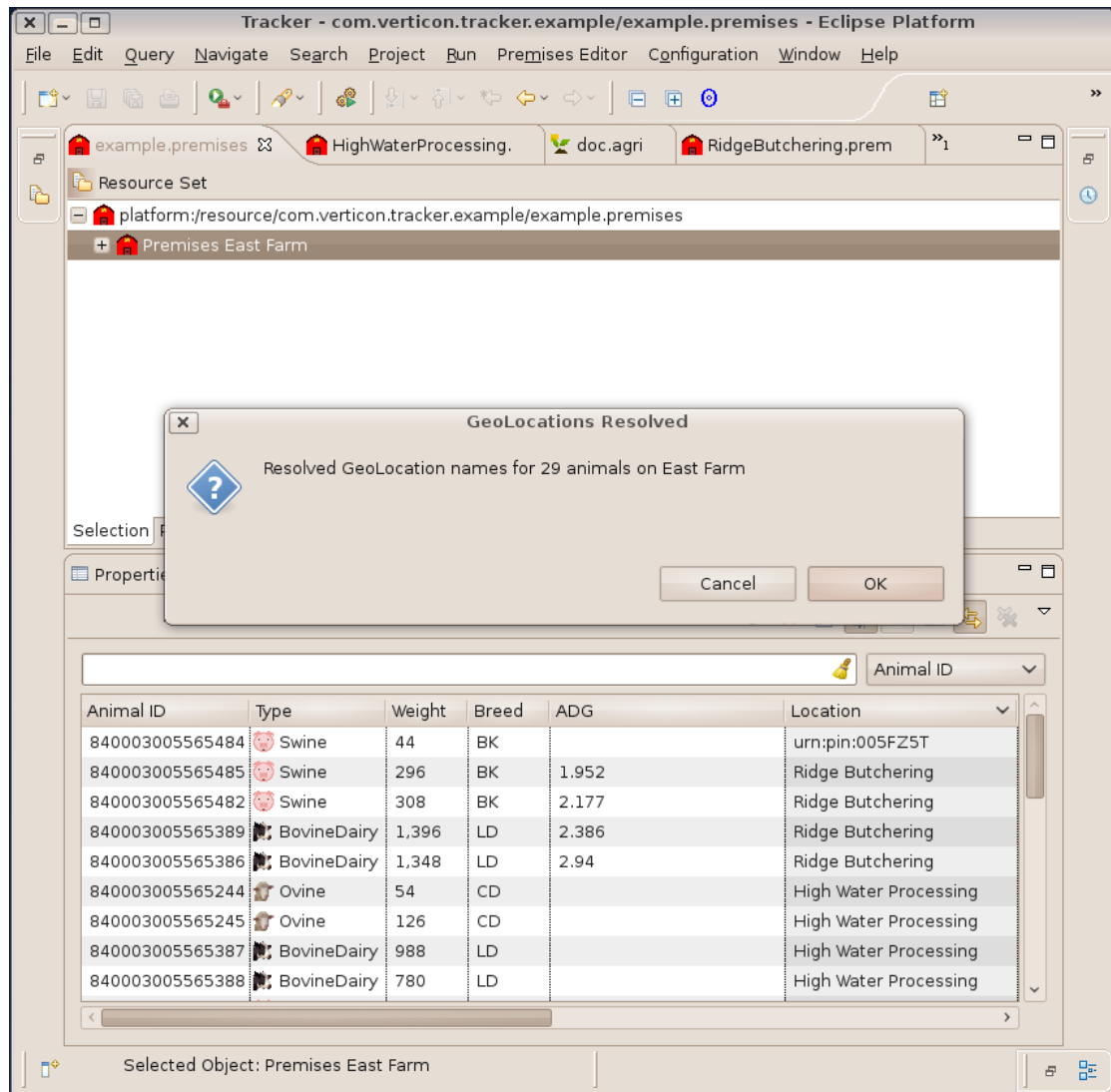
Figure 6.14. Animals View before Geolocation Resolution

Figure 6.15. Animals View after Geolocation Resolution

There are four Animal event types that record Animal (geo)location information.

- MovedIn Animal events record the date, time and URI of the foreign Premises Animals have come from before arriving on Premises.
- MovedOut Animal events record the date, time, and URI of the foreign Premises Animals have been shipped to from the current Premises.
- Position Animal events record the date, time and Geographic coordinates of an Animal.
- Sighting events record the date, time and the name of a location within the current Premises.

In addition to these event types, all events contain a 'pid' attribute for documenting the URI of the Premises that recorded the event. This 'pid' attribute is essential when combining Events from an Animal that was recorded by another premises.

TrackerStore Services

TrackerStore service is a storage and query solution for livestock market exchanges, livestock producers, livestock infrastructure providers and government animal health regulators who need to:

- record animal life and history event information;
- register Livestock Premises geolocation details;
- retrieve complete animal movement histories to trace-back and trace-forward animal movements within and between livestock premises;
- access premises and animal information with state of the art business intelligence analysis and reporting tools;
- share information between collaborating business partners;
- view animal life data and event histories as animals are moved in to a Premises.

The user interacts with TrackerStore Services through Premises Action Commands¹² that are presented as pulldown menus on the Premises Element of Premises documents and on resource folders in a workspace project. The Action commands:

- on the Premises element is a single action command that will record all the Animal Histories in the Premises document to the Tracker Store¹³
- on resource folders there are two action commands, the first for importing documents containing a single animal¹⁴ the second for importing the group of animals that were resident in the premises within a date range¹⁵ from the TrackerStore into in the selected folder¹⁶.

TrackerStore Service also provides functionality for dynamically importing a new Animal with its lifedata and event history from a TrackerStore into an active premises document when its RFID tag is read by a Reader.¹⁷

Bulk Event Processing

Bulk Loading New Events and Animals

Although Premises Editors make it easy to add individual animals and events to Premises Documents, individually adding dozens of animals, or hundreds of events to a Premises is not practical. For bulk loading of animals and events, Premises Editors use Animal Document Templates, Tag IdNumber files and event manipulation wizards to simplify the process.

Like Premises Documents, Animal Document Templates are document files created by Premises Editor Document Creation Wizards. Think of Animal Documents as containing some but not all the information that is in a Premises Document. Specifically a Animal Document is a Premises Document that only contains

¹²Tracker Store Premises Action Commands will only be presented to a user when a TrackerStore Service is configured and accessible by the user.

¹³See Record Animal History to TrackerStore Premises Action Command.

¹⁴To retrieve a single animal the action prompts for an animal id, then creates a document with the animal and all its recorded event histories. The premises in the document will be the last recorded premises that animal resided in.

¹⁵To retrieve a group of animals, the action prompts for a date range and premises uri. The retrieval request will then import from the TrackerStore a premises document containing animals that were resident during the given date range. If no date range is given, just the premises information will be created in the document.

¹⁶See Import Animal from TrackerStore and Import Premises from TrackerStore Premises Action Commands.

¹⁷This requires that the document is being edited in a *Premises Transaction Editor* and that the Animal does not already exist in the document.

a single animal. Like an animal in the Premises Document, an Animal Document Template can contain a tag with an associated set of events. Animal Document Templates serve as a way to represent an event or a set of events that can then be applied to one or more specified animals in a Premises Document.

Within a Premises Document you can specify which animals to apply the events contained in the Animal Document Template by:

1. Directly selecting the animal or a set of animals in the Premises Document with mouse selections.

Tip

One or more animals in the Premises Document can also be selected with OCL Query operations. See *OCL Query Feature* for details.

2. Indirectly selecting an animal or set of animals in the Premises Document from lists of Tag/@idNumbers contained in Tag IdNumber files. See the section called “Tag IdNumber Files”

Animal Document Templates

Animal Document Templates are files that representative a single animal and a set of associated events. Tracker event modification wizards¹⁸ and Animal Document Templates allow the Tracker user to add various combinations of animals and/or events to Premises Documents.

Animal Document Templates have the following characteristics;

- a. Animal Document Templates are persisted as file resources with a *.animal extension;
- b. Animal Document Templates contain a default Animal, a Tag and a set of representative events;
- c. If a Tag is included as a child element of the Animal Element, the Tag/@idNumber is not considered significant and is ignored;
- d. Representative Events in Animal Document Templates are significant and will be copied to specified Animal or set Animals in the Premises Document;
- e. Date Time attributes of Animal Document Templates Events are considered significant dates, if the Date Time value is a date after the year 2000. These attributes are copied to the Event Elements that are added to the Premises Document.

When using tagIdNumber Files with date encoded names, Date Time attributes of Animal Document Templates Events are considered reference dates if the Date Time value of the Templated Event is a date before the year 2000. These attributes are changed to the date encoded in the name of the tagIdNumber file before being copied to the Event Elements that are added to the Premises Document.

Note

When Events are added to Premises Documents using Tracker Readers feature, the Date Time attributes of the Events added to the Premises Document will always have the Date Time attributes set to the current time.

Tag IdNumber Files

Commercial tag readers typically transmit animal id numbers to computers as ASCII encoded characters (For more information on ASCII, see the Wikipedia entry for *ASCII* [<http://en.wikipedia.org/wiki/>

¹⁸Event modification wizards are accessible through popup menus on specific Tracker document elements and File Resources.

ASCII]). Tag Readers send this data to computers over serial communication ports or Bluetooth network connections.

Note

In the Premises Data Model, the Tag id Numbers are represented as `Tag/@idNumber` attributes. A generic way to handle serial and/or Bluetooth tag id number transmissions is to setup the receiving computer's communication facility to capture this data as text files. Capture files containing ASCII numbers that are saved with a *.tags extension are identified as *Tag IdNumber Files*.

To indicate to Premises Editor that the read time of the tags in a Tag IdNumber file is significant, the name of the tag file may be formatted to reflect that time. To timestamp a Tag IdNumber file use the formatting of `yyMMddHHmmss` , where

yy is the last two digits of the year,
MM is the month,
dd is the day of the month,
HH is the hour of the day in 24 hour clock,
mm is the minute of the hour,
ss is the second

Example 6.1. A Time Stamped Tag IdNumber File Name

060409131111.tags. Specifies the date of all `Tag/@idNumber` reads to have occurred at 2006-04-09T13:11:11.

Note

To use the Time Stamped Tag IdNumber File Names the dates of the events in an Animal Document Template must be set to date times earlier than the 2000 AD reference.

Tip

Tag IdNumber Files can also be utilized by the Reader Feature.

Usage and Operations

Adding Events to Manually Selecting Animals

To add a series of events to an animal or set of animals in a Premises Document select the animal or animals and invoke a popup menu item;

- a. Right click on the Animal to get the top level popup menu;
- b. Select the 'Tracker' menu item to get the Tracker popup menu;
- c. Select 'Add Template Events to Selected Animals' to bring up a selection window to select a Template;
- d. Choose an Animal Document Template containing the set of events you wish to add the animal.

Add Events and/or Animals to a List of Animals Specified in a File

To add a series of events and/or animals to the Premises as a whole, you will still need to specify which animals to add event to. This can be done indirectly with a Tag IdNumber file. Select a Tag IdNumber file and invoke a popup menu item it.

- a. Right click on the Animal Document Template that contains the list of animal ids which specify the set of animals in the Premises Document that you wish to add events to.
- b. From the popup menu Select 'Tracker' to get the Tracker specific menu.
- c. Select 'Add Template Events to Premises' to bring up a selection window that will allow you to select a Animal Document Template.
- d. Choose an Animal Document Template containing the set of template events and the default animal type you wish to add the file specified animals in the premises.

Selecting and Targeting Elements

When working with Premises documents that contain large numbers of animals and events it is often necessary to focus on a subset of the Animals and Events contained within the Premises.

The Target Selection Tool can be used to restrict the contents of the Premises Editor Selection tree, the Animals View, and the Events View and show only Animals and Events that are of interest.

- a. Select one or more animals or events. See Figure 6.16, “Selecting Animal Events ”.
- b. Press the *Target Selection* button to restrict the views and editor to the selected animals and events. See Figure 6.17, “After Invoking Target Selection ”.
- c. Press the *Target Selection* to return to an unrestricted viewing of all the animals and events.

Figure 6.16. Selecting Animal Events

The screenshot shows the Eclipse Platform interface with the Tracker application. The main window is titled "Tracker - com.verticon.tracker.example/example.premises - Eclipse Platform". The menu bar includes File, Edit, Query, Navigate, Search, Project, Run, Premises Editor, Configuration, Window, and Help. The toolbar contains various icons for file operations and navigation.

The left sidebar shows the "Resource Set" view, which displays a tree structure of premises and animals. The tree is expanded to show the "Premises East Farm" and its associated animals. The "Events" view is active, displaying a list of animal events. The "Events" view has a "Target Selections" button and a "Tag ID" dropdown. The events are listed in a table with columns for Date, Event, and Animal.

The bottom sidebar shows the "Animals" view, which displays a table of animal details. The table has columns for Animal ID, Type, Weight, Breed, ADG, Location, BirthDate, Sex, and Last Event. The table is filtered to show only the animals selected in the "Events" view.

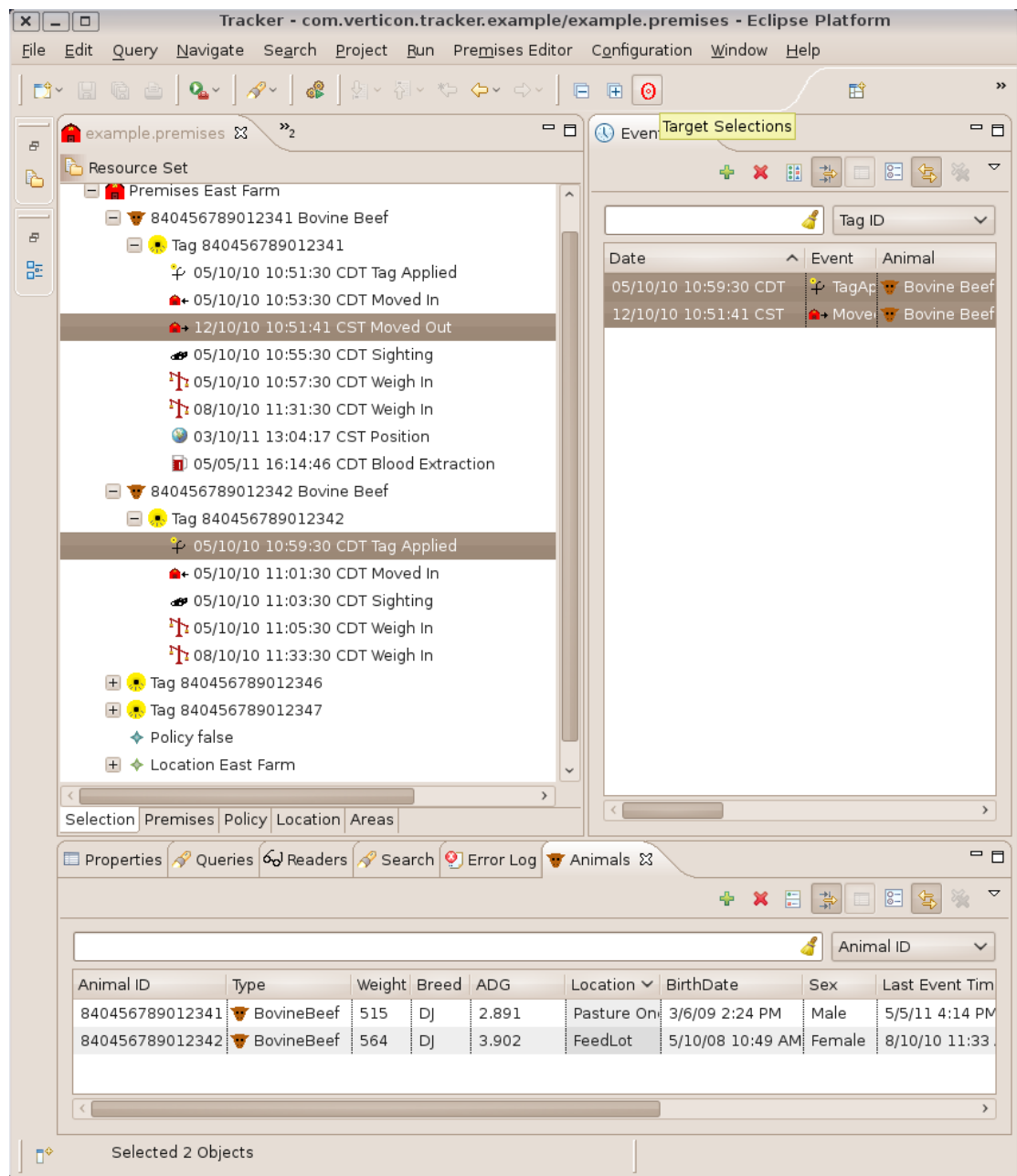
The "Events" view table data is as follows:

| Date | Event | Animal |
|-----------------------|--------|----------|
| 05/10/10 11:15:30 CDT | TagAp | Bovine B |
| 05/10/10 11:17:30 CDT | Move | Bovine B |
| 05/10/10 11:19:30 CDT | Sighti | Bovine B |
| 05/10/10 11:21:30 CDT | Weigh | Bovine B |
| 05/10/10 11:23:30 CDT | TagAp | Bovine B |
| 05/10/10 11:25:30 CDT | Move | Bovine B |
| 05/10/10 11:27:30 CDT | Sighti | Bovine B |
| 05/10/10 11:29:30 CDT | Weigh | Bovine B |
| 08/10/10 11:31:30 CDT | Weigh | Bovine B |
| 08/10/10 11:33:30 CDT | Weigh | Bovine B |
| 08/10/10 11:35:30 CDT | Weigh | Bovine B |
| 08/10/10 11:37:30 CDT | Weigh | Bovine B |
| 08/10/10 11:39:30 CDT | Weigh | Bovine B |
| 12/10/10 10:51:41 CST | Move | Bovine B |
| 03/04/11 11:35:00 CST | Move | Bovine B |
| 03/04/11 14:30:17 CST | Sighti | Bovine B |
| 03/04/11 14:30:17 CST | TagAp | Bovine B |
| 03/10/11 13:04:17 CST | Positi | Bovine B |
| 04/06/11 14:38:09 CDT | Castr | Bovine B |

The "Animals" view table data is as follows:

| Animal ID | Type | Weight | Breed | ADG | Location | BirthDate | Sex | Last Event |
|-----------------|-------------|--------|-------|-------|-------------|-----------|-----|-------------|
| 840003005565486 | Swine | 274 | BK | 1.839 | urn:pin:003 | | | 9/21/09 11: |
| 840003005565385 | BovineDairy | 1,100 | LD | 2.012 | urn:pin:003 | | | 9/21/09 11: |
| 840003005565487 | Swine | 308 | BK | 2.033 | urn:pin:003 | | | 9/21/09 11: |

The status bar at the bottom indicates "Selected 2 Objects".

Figure 6.17. After Invoking Target Selection

Premises Action Commands

For key workspace resources and elements within the Premises document, the Premises Editor provides action commands that the user can invoke to effect changes on the resource or document element.

Tracker Actions can be invoked on specific types of workspace resources. These actions are accessible when viewing resources with the Project Explorer View.¹⁹

¹⁹In the Project Explorer View right click on the resource and the popup main menu will be shown. From there the sub menu 'Tracker Actions' will contain the appropriate action command.

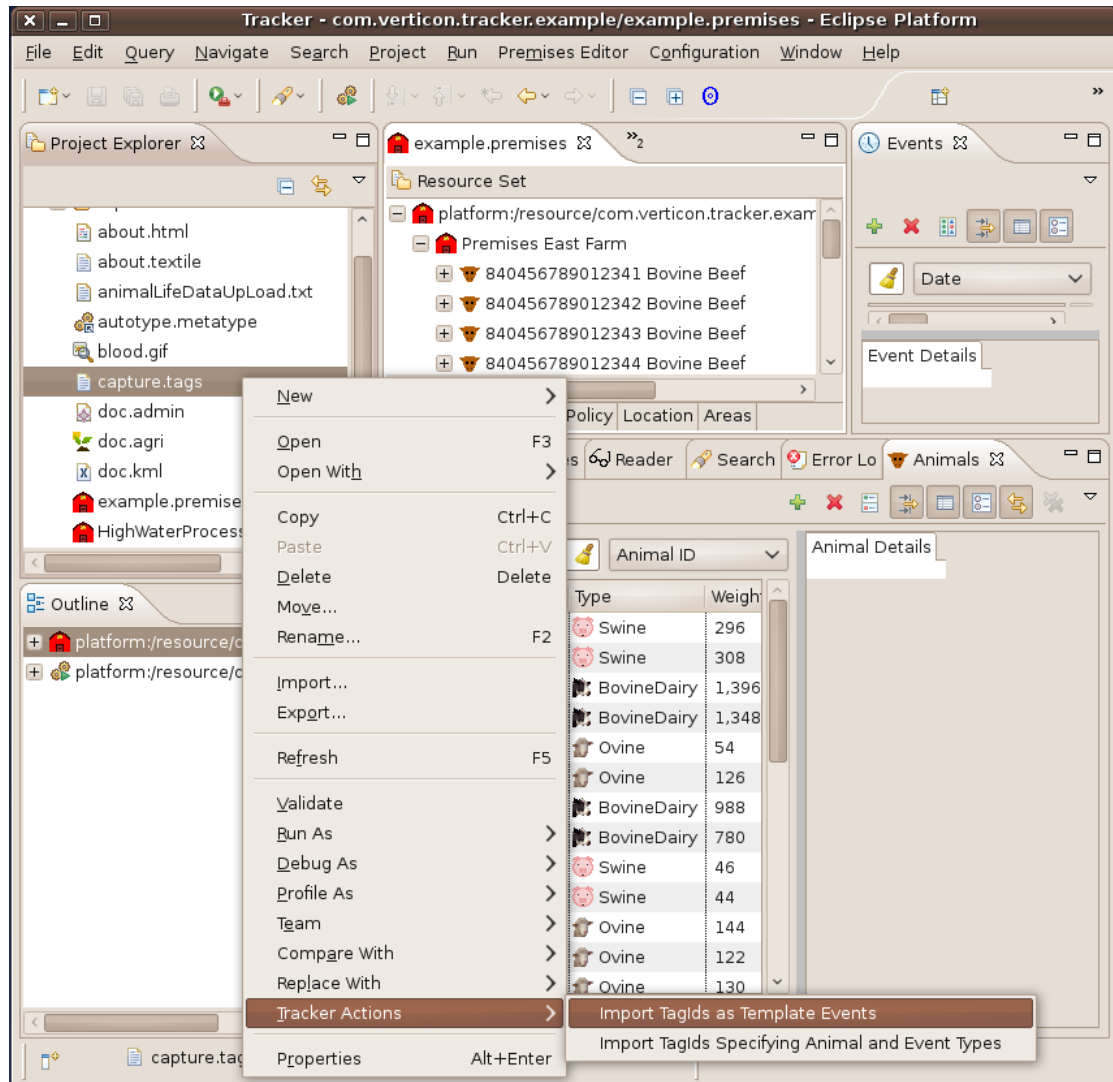
Figure 6.18. Invoking Resource Actions from Project Explorer

Table 6.2. File Resource Action Commands

| Resource | Type and Format | Command Name | Description | Comment |
|----------|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------|
| *.tags | Text files with extension 'tags', that contain a list of tag ID numbers. ^a | Import TagIds as Template Events | Adds events from the template to animals. ^b | Adds multiple events to existing animals or adds new animals with a common life data. |
| *.tags | Text files with extension 'tags', that contain a list of tag ID numbers. | Import TagIds Specifying Animal and Event Types | Adds the selected event to animals in the document. ^c | Adds single events to multiple existing animals or adds new animals with a common life data. |
| *.txt | Comma separated value CSV text files with extension 'txt', contain Animal life data for a set of animals. ^d | Import Text File as Animal Life Data | Adds animals with life data to the document. ^e | Use to import new animals with unique life data. |
| *.log | Comma separated value CSV text files with extension 'log', contain Animal event data for a set of animals in <i>Name Value</i> pairs. ^f | Import Log File as Animal Event Data ^g | This is one of the formats for logging output by the iRouter logger feature. | Log files use an easy to read format. ^h |
| *.csv | Register premises information from a csv file to TrackerStore. ⁱ | Register Premises | Required columns ^j | Use to register multiple premises when data can be exported from an existing datasource. |

^a Each number is on its own line and lines are terminated with a carriage return. These text files can be simply created in a text editor by hand or exported from other applications and data sources. Most data management applications have text file export utilities.

^b Prompts for an Animal template. Each tagId number is mapped to the Animal and the Events in the template. The Events in the template will be given the current time and added to the Animal that corresponds with the TagId. If no Animal exists with the tagId, a new Animal will be created with the type and attributes of the template Animal.

^c Prompts for Animal and Event types. Event will be given the current time and added to the Animal that corresponds with the TagId. If no Animal with the tagId exists, a new Animal will be created.

^d Values are separated with commas and rows are on lines terminated with a carriage return. These files can be created in text editors or spreadsheet application and exported as csv file.

^e Prompts to map column values to Animal feature names. Two essential Animal feature must be mapped, Animal.id and Animal.species. Next page will map the values in the selected species columns to Premises Animal species names.

^f Name value pairs are separated with a comma and each row is on lines terminated with a carriage return. Example: type='measurement', dateTime='2010-02-22 10:23:46', id='900014000554891', scope='animal.weight', value='81.0', error='0.0100', unit='kg'

^g Only available when the iRouter Premises Gateway feature is provisioned.

^h While these files are useful for adding event data, other programs may not be able to easily output such a format.

ⁱ Each value is quote enclosed and separated with a comma. Each line is terminated with a carriage return. These text files can be simply created in a text editor by hand or exported from other applications and data sources. Most data management applications have text file export utilities.

This command will not be visible if there are no TrackerStore Services configured.

^j Column values used for mapping: Pin, Name, Phone, Email, Street, City, State, zipCode, latitude, longitude

Table 6.3. Folder Resource Action Commands

| Resource | Type and Format | Command Name | Description | Comment |
|----------|-------------------------------|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| folder | A file directory ^a | Import Animal from Tracker Store ^b | User will first be prompted to select which Tracker Store Server to use to retrieve the Animal, then for an Animal id. | Imports into the selected directory a premises document containing the animal and any metatypes associated with the animal. |
| folder | A file directory ^c | Import Premises from Tracker Store ^d | User will first be prompted to select which Tracker Store Server to use to retrieve the Premises, then for a Premises id and for a range of dates. | Imports into the selected directory a premises document containing all the animals that resided in the premises within the date range and any metatypes associated with the animals. If no dates are given than the premises without any animals is imported. |

^aAlso known as a file folder

^bThis command is also available as an Import Wizard: *Import an Animal from the TrackerStore into the workspace.*

This command will not be visible if there are no TrackerStore Services configured.

^cAlso known as a file folder

^dThis command is also available as an Import Wizard: *Import Premises from the TrackerStore into the workspace.*

This command will not be visible if there are no TrackerStore Services configured.

Tracker Actions can be invoked on Premises document elements. These actions may be accessed as popup menus on the document elements when viewing documents from the Selection page of the Premises Editor, document tree in the Outline View tree, the master table in the Animals View or the master table in the Events View.²⁰

²⁰ Right click on the element and the popup main menu will be shown. From there the sub menu 'Tracker Actions' will contain the appropriate action command.

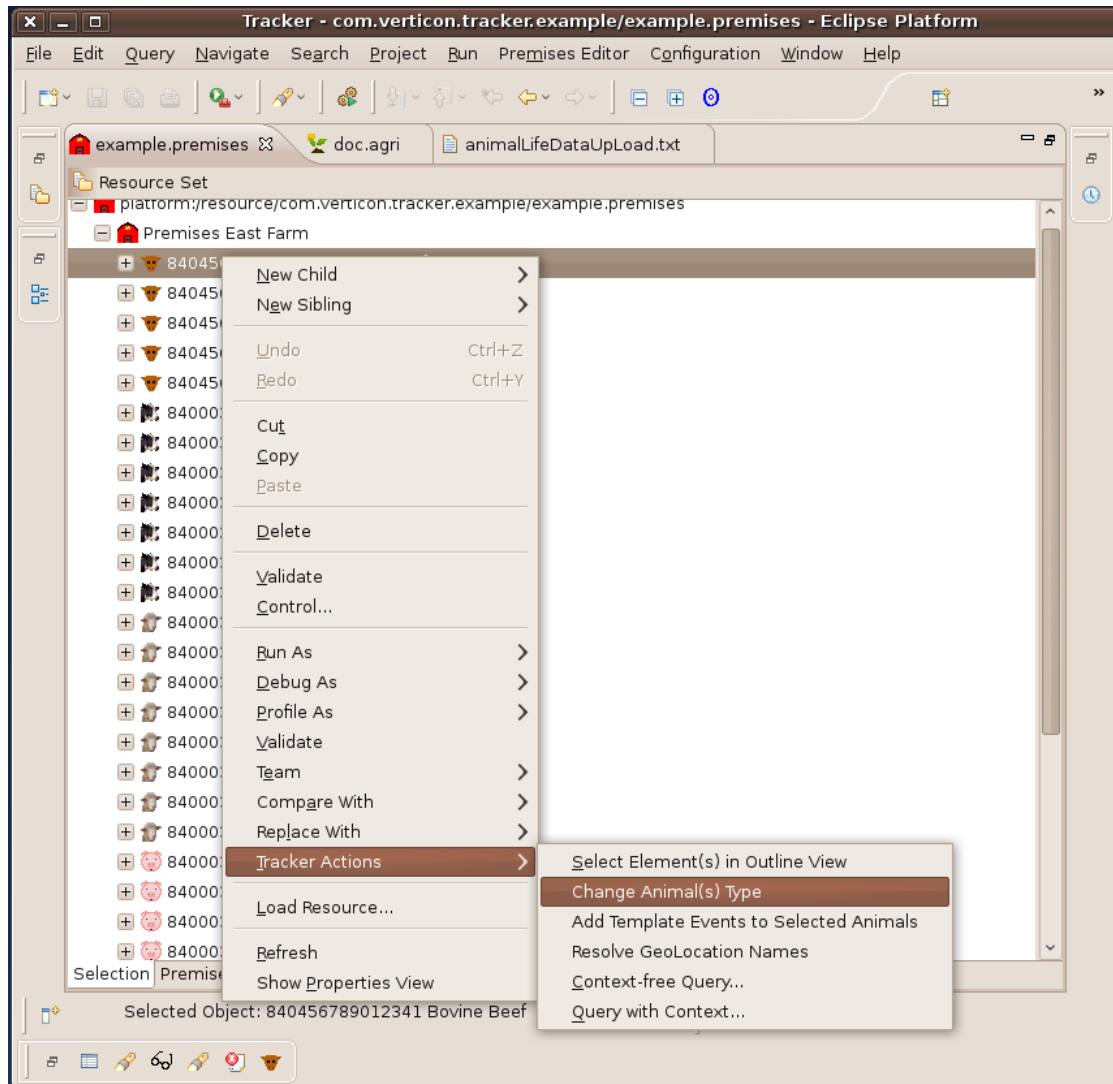
Figure 6.19. Invoking Element Actions from Premises Editor Selection Page

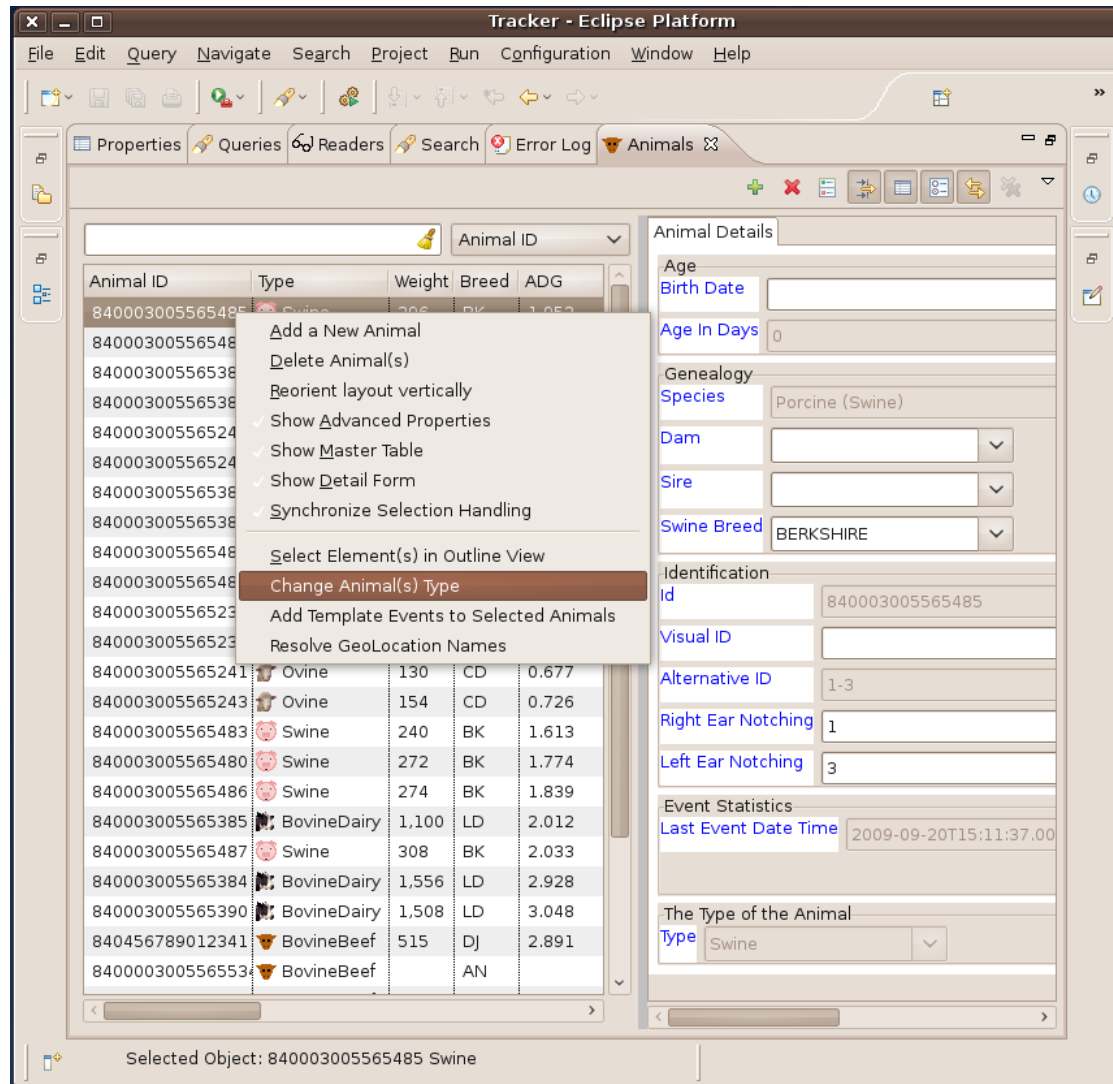
Figure 6.20. Invoking Element Actions from Animals View Master Table

Table 6.4. Action Commands on all Elements

| Element | Command Name | From Selection Page ^a | From Animals View | From Events View | Description | Comment |
|--------------|-----------------------------------|----------------------------------|-------------------|------------------|--------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------|
| All elements | Select Element(s) in Outline View | Yes | Yes | Yes | Selects the same element or elements in the tree of the Outline View. | Helpful to sync from Views to the context of a tree view. |
| All elements | Context Free Query ^b | Yes | No | No | Shows a dialog for entry of a query to select sub (child) elements of the selection. | From a Premises element enter a query to find animals or events. ^c |
| All elements | Query with Context ^d | Yes | No | No | Shows a dialog for entry of a query to select sub (child) elements of the selection. A combo box for selection of the query context. | Select the Premises and query for animals or events. |

^aThe Outline View like the Selection Page of the Premises Editor shows the document as a tree. All actions that can be accessed as popup menus in the Selection Page are also available from the Outline View.

^bOnly available when the OCL feature is provisioned.

^cSee the OCL Query documentation for more details.

^dOnly available when the OCL feature is provisioned.

Table 6.5. Action Commands on Animal Elements

| Element | Command Name | From Selection Page^a | From Animals View | From Events View | Description | Comment |
|----------------|-----------------------------------------|----------------------------------------|--------------------------|-------------------------|----------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
| Animal(s) | Change Animal(s) type | Yes | Yes | No | Prompts to select a new animal type. | Use this when importing animals and not all of them are the same type. ^b |
| Animal(s) | Add Template Events to Selected Animals | Yes | Yes | No | Prompts the user to select a template event. | Use this to add events from an Animal template to existing Animals. |
| Animal(s) | Resolve Geolocation Names | Yes | Yes | No | Resolves all geolocation names on the Animal(s) and the events of the Animal(s). | Use this for traceback and traceforward of Animals. ^c |

^aThe Outline View like the Selection Page of the Premises Editor shows the document as a tree. All actions that can be accessed as popup menus in the Selection Page are also available from the Outline View.

^bFirst import them as the same type with one of the Import actions, then select the ones to change and invoke this command.

^cResolves the urns on Moved events to real names, geo coordinates on Position events to real names and for resolving the current location of the Animal(s) as a location name.

Table 6.6. Action Commands on Event Elements

| Element | Command Name | From Selection Page ^a | From Animals View | From Events View | Description | Comment |
|---------------------------|---------------------------------------|----------------------------------|-------------------|------------------|--------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------|
| Event(s) | Modify Selected Events | Yes | No | Yes | Modifies one or more of the common attributes of the selected Event(s). | Sets the comments, the correction and the electronically read flags for more than one event. |
| MovedIn or MovedOut Event | Set Foreign Premises URI ^b | Yes | No | Yes | Use the Location Service to assist in setting MovedIn and MovedOut event URI and pin attributes. | Work with names instead of numbers when setting MovedIn and MovedOut uri and pin attributes. ^c |

^aThe Outline View like the Selection Page of the Premises Editor shows the document as a tree. All actions that can be accessed as popup menus in the Selection Page are also available from the Outline View.

^bThis command will not be visible when Location Services are not configured.

^cFirst configure the Location Service with business partners. Next when setting attributes on Moved events invoke this command to set the URI of the foreign Premises. This command will present the user with a dialog containing the names and the URIs of all the partners from the Location Service. Choose one to set the URI of the Moved event. If the chosen partner has an URI that is a USDA Premises ID number, the MovedIn or MovedOut pin attribute will be set as well.

Table 6.7. Action Commands on Premises Elements

| Element | Command Name | From Selection Page ^a | From Animals View | From Events View | Description | Comment |
|----------|----------------------------------------------------|----------------------------------|-------------------|------------------|---------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|
| Premises | Resolve Geolocation Names ^b | Yes | No | No | Resolves all geolocation names on all Animals and Events in the Premises. | Use for traceback and traceforward of all Animals. ^c |
| Premises | Register Premises to TrackerStore ^d | Yes | No | No | Registers (stores) the Premises to TrackerStore | Administrator command to register a premises. |
| Premises | Record Animal History to TrackerStore ^e | Yes | No | No | Publishes (stores) Animal Life Data and Event Histories to TrackerStore | The user designated as the publisher of the Premises will use this command to record Animal Life Data and Event Histories ^f . |
| Premises | Record Animal History to TrackerStore ^g | Yes | No | No | Publishes (stores) Animal Life Data and Event Histories to MongoDB | The user designated as the publisher of the Premises will use this command to record Animal Life Data and Event Histories ^h . |

^aThe Outline View like the Selection Page of the Premises Editor shows the document as a tree. All actions that can be accessed as popup menus in the Selection Page are also available from the Outline View.

^bCommand will not be visible if there are no Location Services configured.

^cResolves the urns on Moved events to real names, geo coordinates on Position events to real names and for resolving the current location of the Animal(s) as a location name.

^dThis command will not be visible if there are no TrackerStore Services configured and the user is not a Server Administrator.

^eThis command will not be visible if there are no TrackerStore Services configured and the user is not a Premises Publisher.

^fWhen animal event histories are recorded, the system will record a time stamp. This time stamp is used to insure that only the latest animal history events are recorded and changes made to old events are not.

^gThis command will not be visible if there are no TrackerStore Services configured and the user is not a Premises Publisher.

^hWhen animal event histories are recorded, the system will record a time stamp. This time stamp is used to insure that only the latest animal history events are recorded and changes made to old events are not.

Premises Data Model

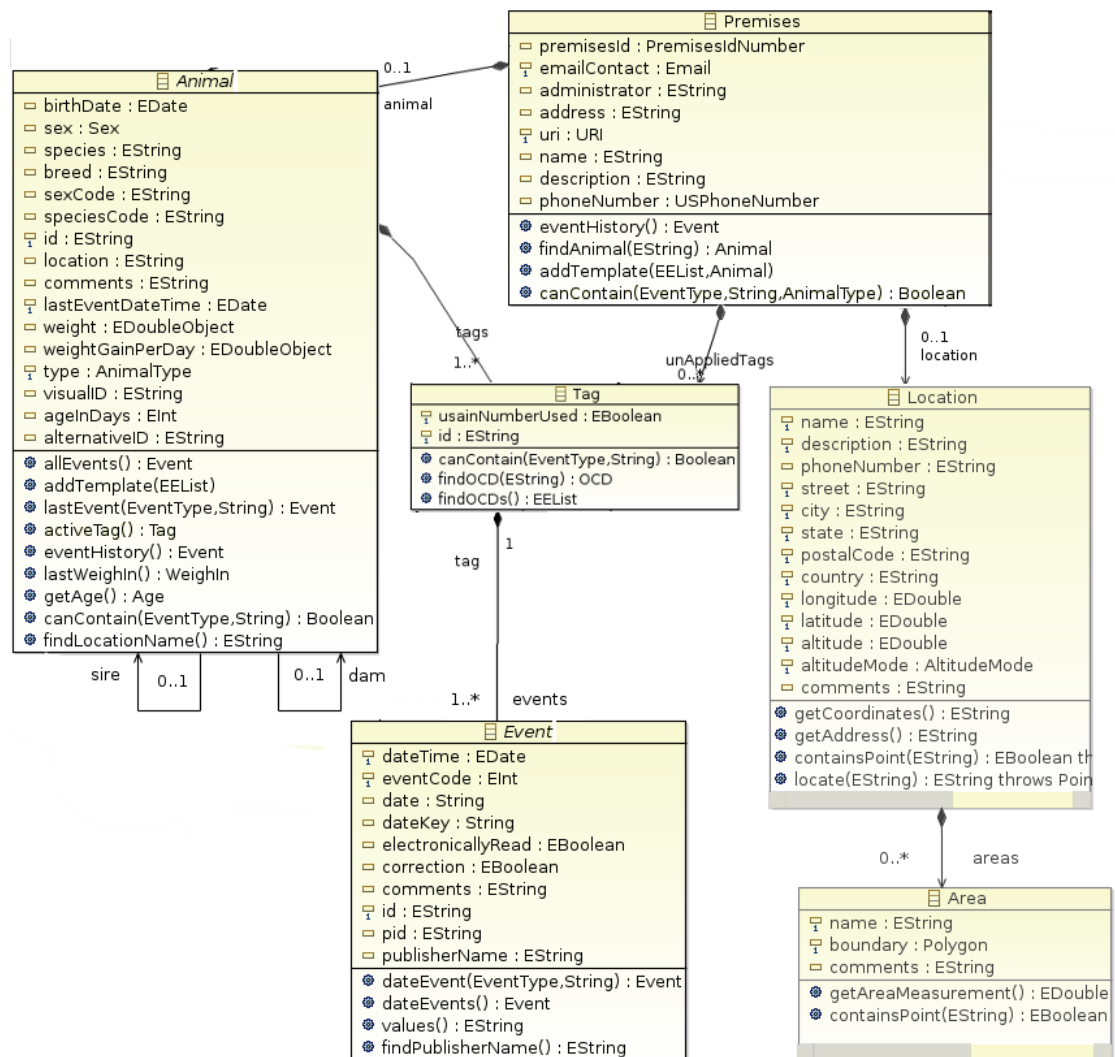
Because all user activity centers around the creation and manipulation of *Premises Documents*, understanding the structure (called the *data model*) of Premises documents is the easiest way to learn to use the information management facilities of the Premises Editor.

Data Model Core Elements

All Premises documents structurally conform to the *Premises Data Model*. In the Premises Data Model, a single Premises contains zero or more animals. Each Animal must have at least one but may have more than one (identification) tag. Each Tag must have at least one but may have more than one associated event.

Core elements of the Premises Data Model are shown in Figure 6.21, “Premises Data Model Core Elements”.

Figure 6.21. Premises Data Model Core Elements



- a. The core elements of the Premises Model are depicted as rectangles in the figure.
- b. The notation within the elements are the attributes of the element.
- c. The notation between the elements are the relationships of the elements.

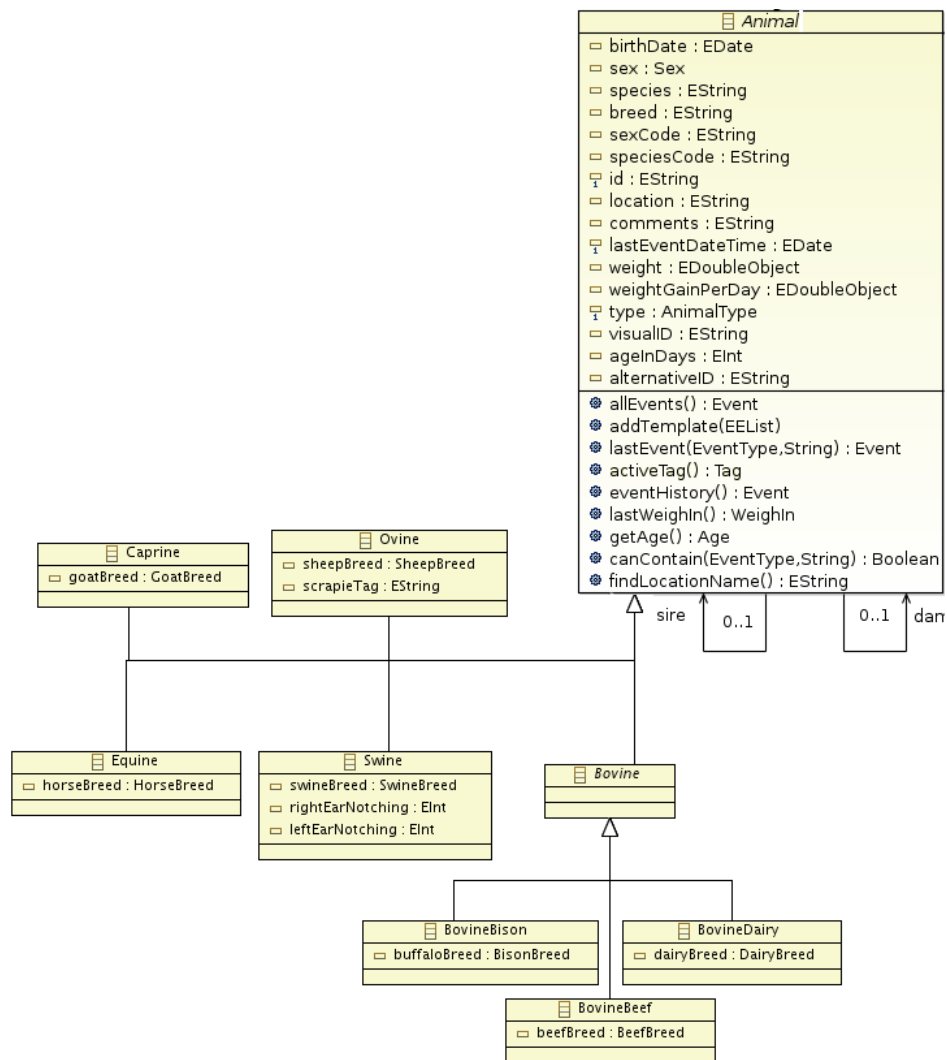
The following table offers a text explanation of the primary element relationships.

Table 6.8. Core Elements of the Premises Model

| Element Name | Relationship to other Elements |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Premises | There is one premises element in a Tracker document. It is consider the root of the Premises document. All other elements are children elements of the Premises element. The Premises Editor tab 'Premises' provides a form for editing the attributes of the Premise element. |
| Location | A premises can have an optional Location element to specify address and geolocation details. The Premises Editor tab 'Location' provides a form for editing the attributes of the 'Location' element. |
| Area | A 'Locaton' element can have zero or more optional 'Area' elements to specify sublocation boundries. Areas specify locations within the Premises. Specifying areas in a premises allow animals with 'Sighting' and 'Position' events to be located and traced as they move within the premises. The Premises Editor tab 'Areas' provides a form for creating and editing the attributes of one or more 'Area' elements. |
| Animal | A premises can have zero or many animals. |
| Tag (assigned) | An Animal can have one or many tags. |
| Tag (unassigned) | A Premises can have zero or many tags that are unassigned. |
| Event | A Tag can have one or many events. |
| Policy (not shown) | A Premises can have zero or one Policy for restricting the premises to specific animal species and event types. The Premises Editor tab 'Policy' provides a form for creating and editing the attributes of the 'Policy' element. |

Data Model Animal Elements

The Premises Data Model supports multiple Animal types as shown in Figure 6.22, “Premises Data Model Animal Elements”

Figure 6.22. Premises Data Model Animal Elements

Data Model Event Elements

The Premises Data Model supports multiple Event types as shown in Figure 6.23, “Premises Data Model Event Elements (part 1)” and Figure 6.24, “Premises Data Model Event Elements (part 2)”.

Figure 6.23. Premises Data Model Event Elements (part 1)

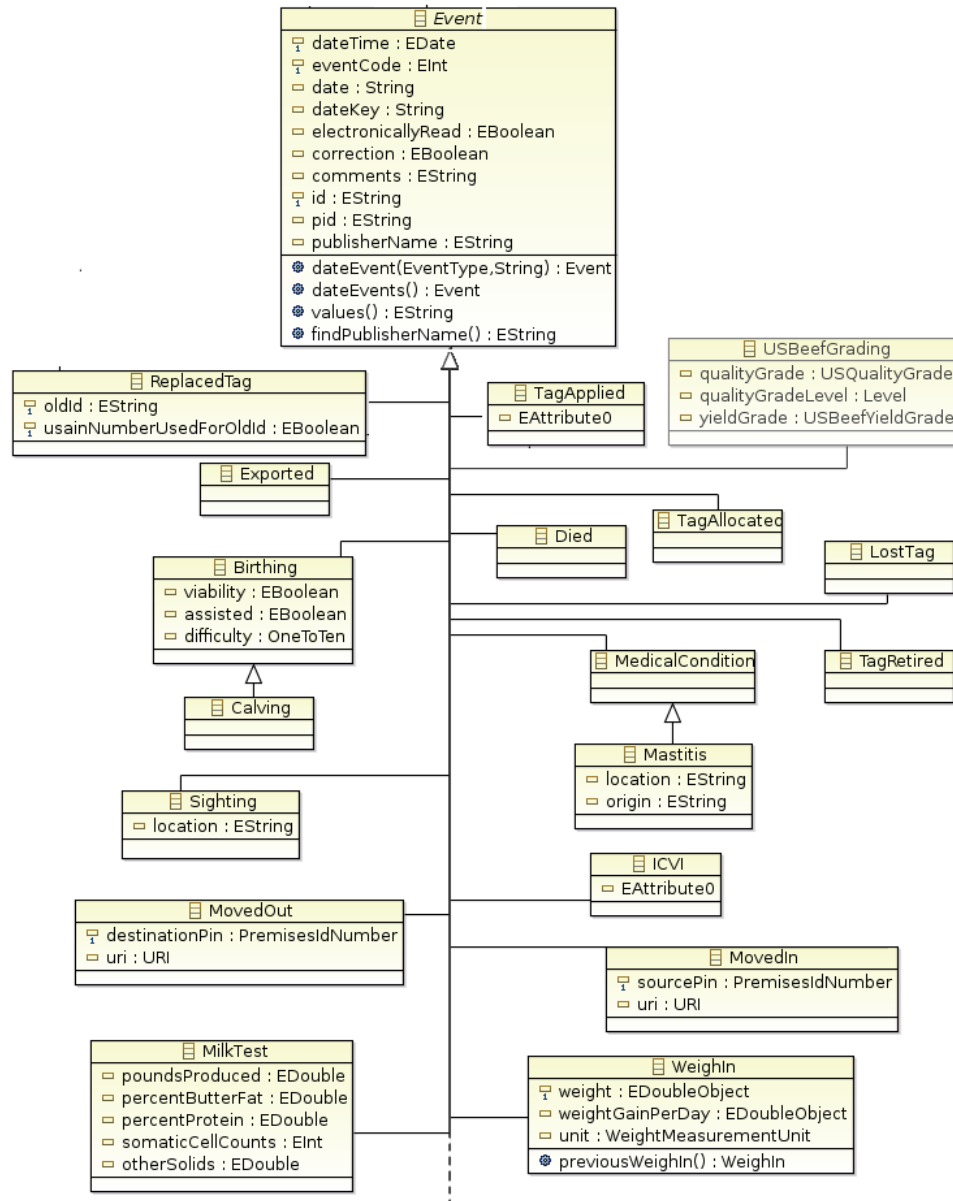
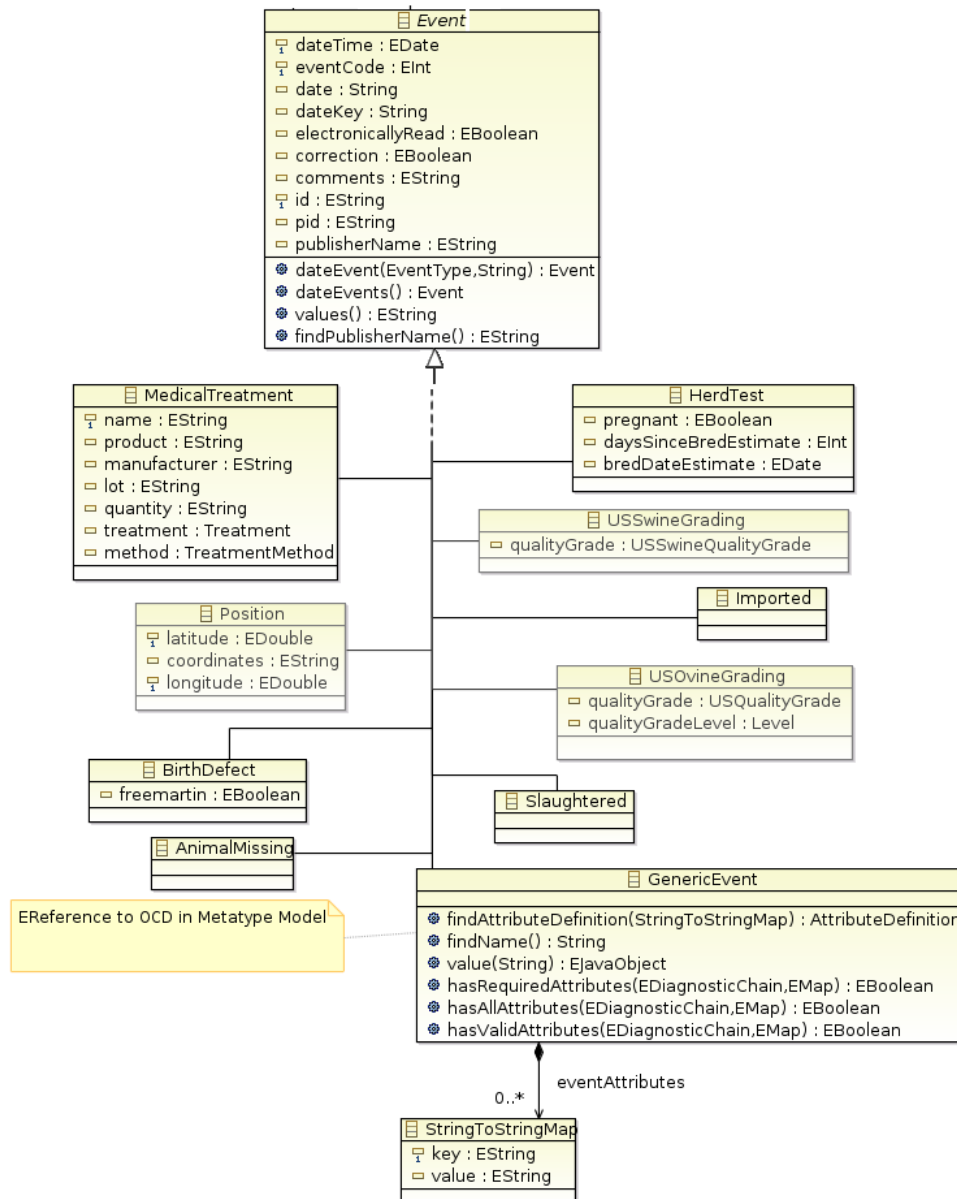


Figure 6.24. Premises Data Model Event Elements (part 2)

Custom Events

The Premises Editor ships with predefined event types that are suitable for documenting most livestock histories. If the predefined event types are not sufficient, the Premises Editor utilizes the *Metatype Editor Feature* and the Metatype documents it creates to define custom event types.

Creating Custom Events

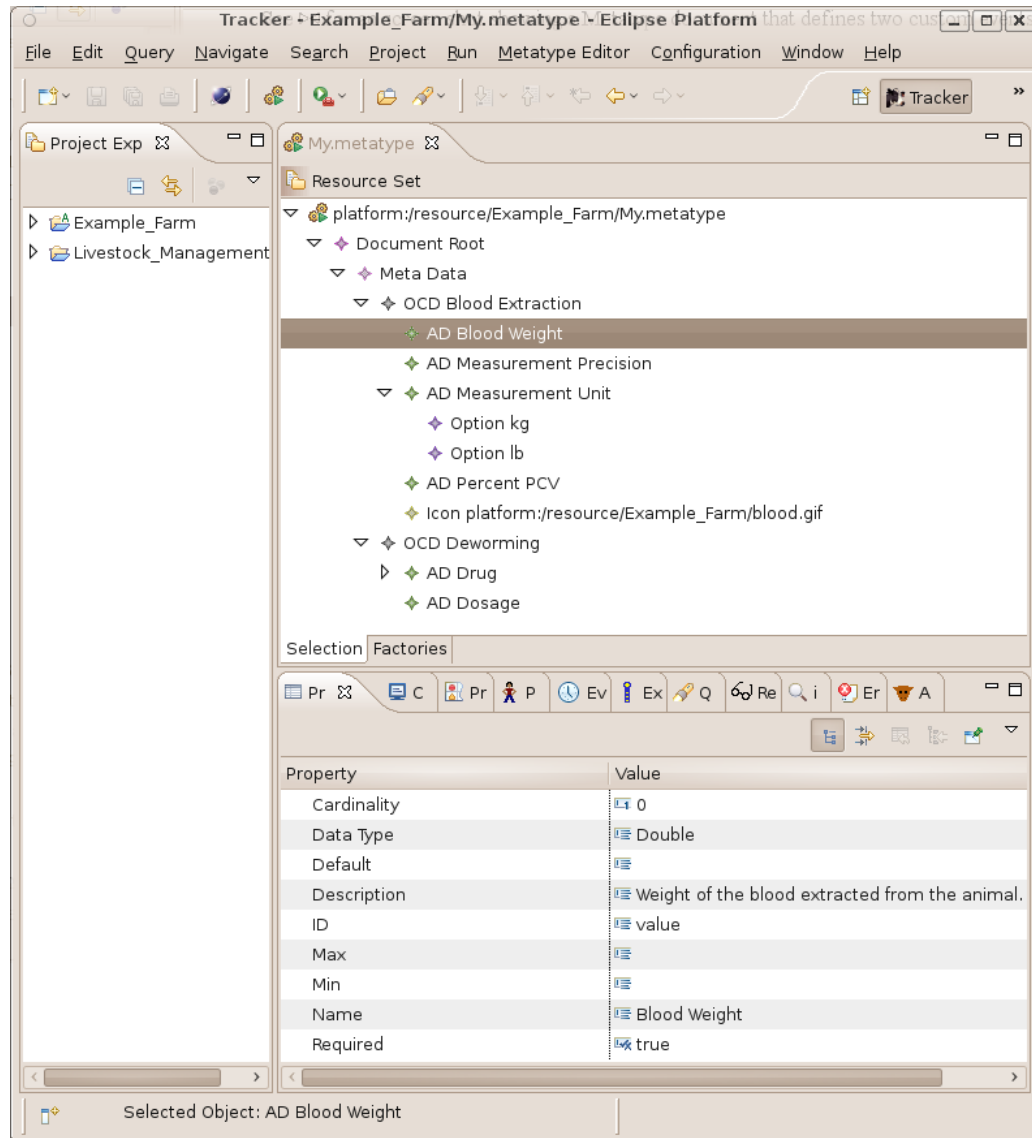
To define one or more custom events:

- Create a Metatype document using the **New Metatype Document Wizard**

- For every custom event define an *Object Class Definition*, OCD element; and in the *Properties view* specify its properties as:
 - *Description* - an optional description of the custom event
 - *ID* - a unique string identifying the event within the document
 - *Name* - a user recognizable name for the event
- Within each OCD create one or more child *Attribute Definition*, AD elements to define the attributes of the custom event. These attributes will be displayed in the editor as properties of the event.
- Within each AD child element specify the properties:
 - *Cardinality*, an optional property that specifies the number of values contained in this attribute. The default is 0.
 - *Data Type*, a required property that specifies what type of property value is valid for the attribute. A pull down list will show the options: *String*, *Long*, *Integer*, *Short*, *Char*, *Byte*, *Double*, *Float* and *Boolean*

When working with custom events in premises documents only values appropriate to the specified data type will be considered valid. For example a custom event with a data type of *Boolean* will only accept *true* and *false* as valid values.
 - *Default*, is the default value for this attribute. This default value(s) must be appropriate for the cardinality and data type defined above.
 - *Description*, an optional description of the attribute. This will be shown in the message panel when the user interface is focused on this attribute.
 - *ID*, document wide identification string for this attribute.
 - *Max*, maximum value for this attribute.
 - *Min*, minimum value for this attribute.
 - *Name*, user recognizable name for this attribute.
 - *Required*, whether this attribute must be given a value by the user for the event to be considered valid.
- Within each OCD optionally create a child *Icon* element to associate an image with the event.

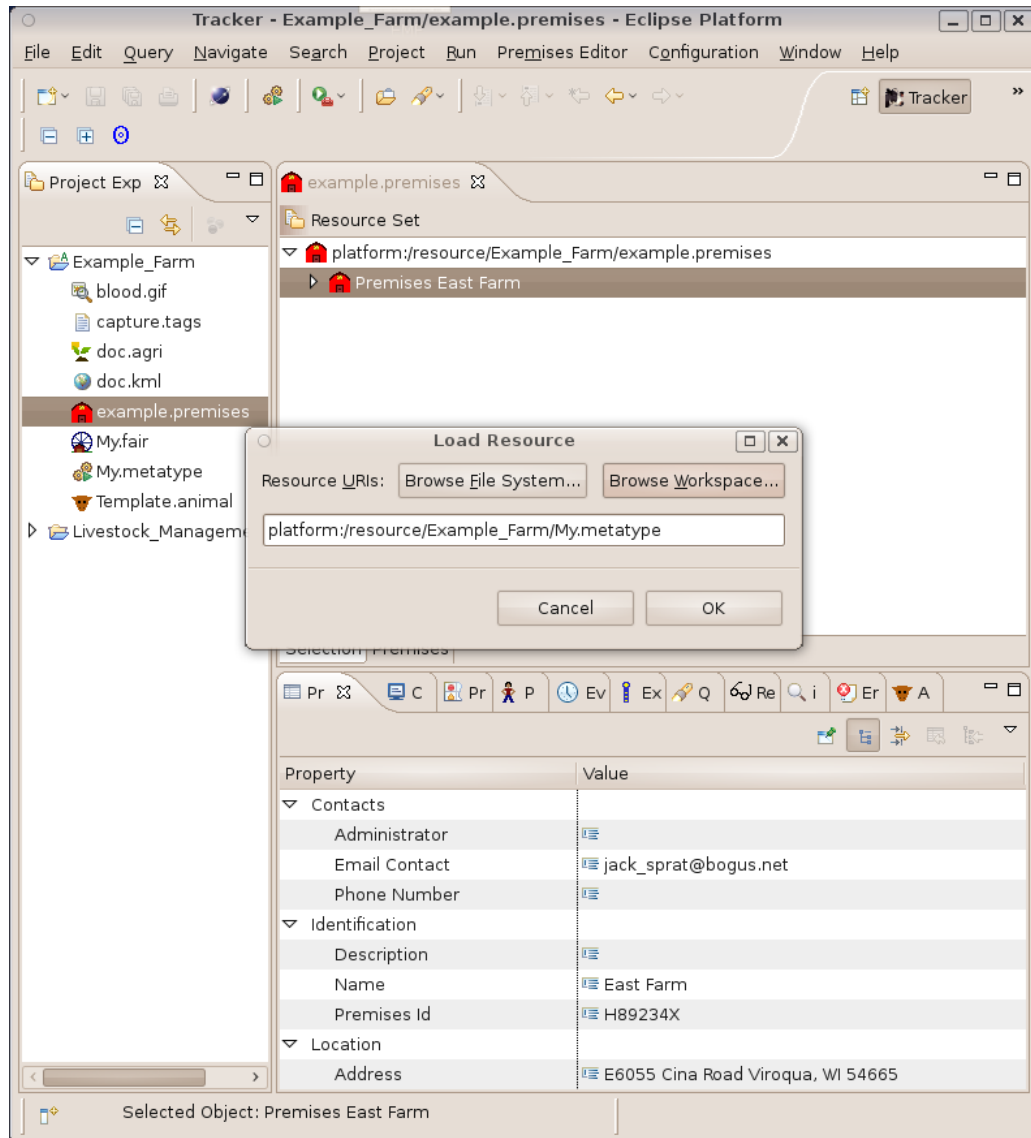
See Figure 6.25, “Defining Custom Events” for an screen shot showing a Metatype document that defines two custom events.

Figure 6.25. Defining Custom Events

Using Custom Events in Premises Documents

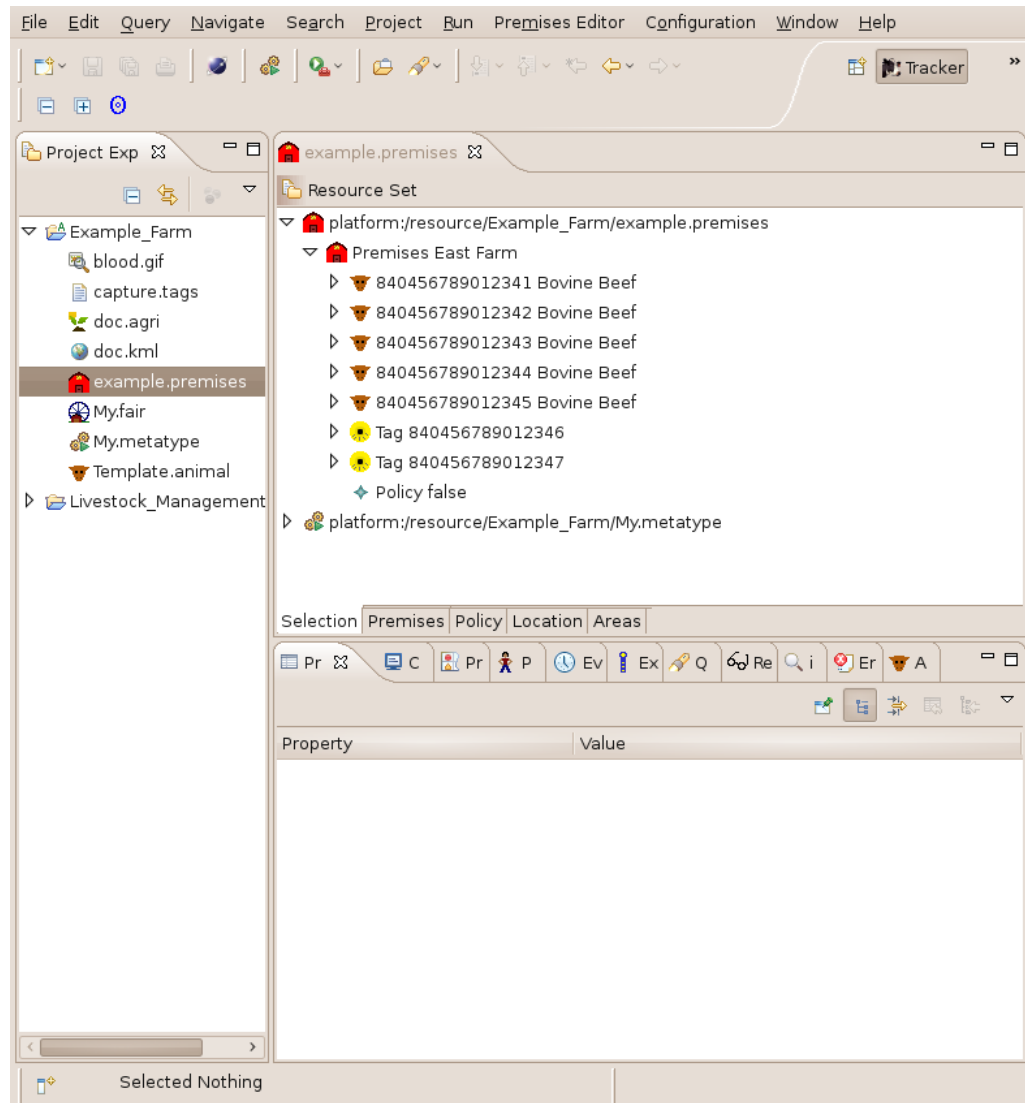
To use custom events in Premises documents, the Metatype document that defines the custom events must first be loaded as a resource.²¹ To load a resource into a Premises document place the mouse cursor on the root element in the Premises document, right click, and from the pull down menu choose Load Resource...

²¹This only needs to be done once. When the premises document is saved and opened again, the Metatype document with the custom event specifications will automatically be associated with the Premises document.

Figure 6.26. Load Resource Dialog

From the Figure 6.26, “Load Resource Dialog” choose the Metatype document that defines the custom events and press OK.

See Figure 6.27, “Premises Document Loaded with a Metatype Resource”.

Figure 6.27. Premises Document Loaded with a Metatype Resource

When a Premises document is loaded with a Metatype resource, all the custom events from the Metatype document are available for adding to the animal histories of the Premises Document. The Premises Editor user interfaces will also treat the Metatype documents custom events just like predefined events.

See Figure 6.28, “Adding an Event in the Premises Editor”, Figure 6.29, “Adding and Event in the Event View”, and Figure 6.30, “Modifying Attributes in the Events View”.

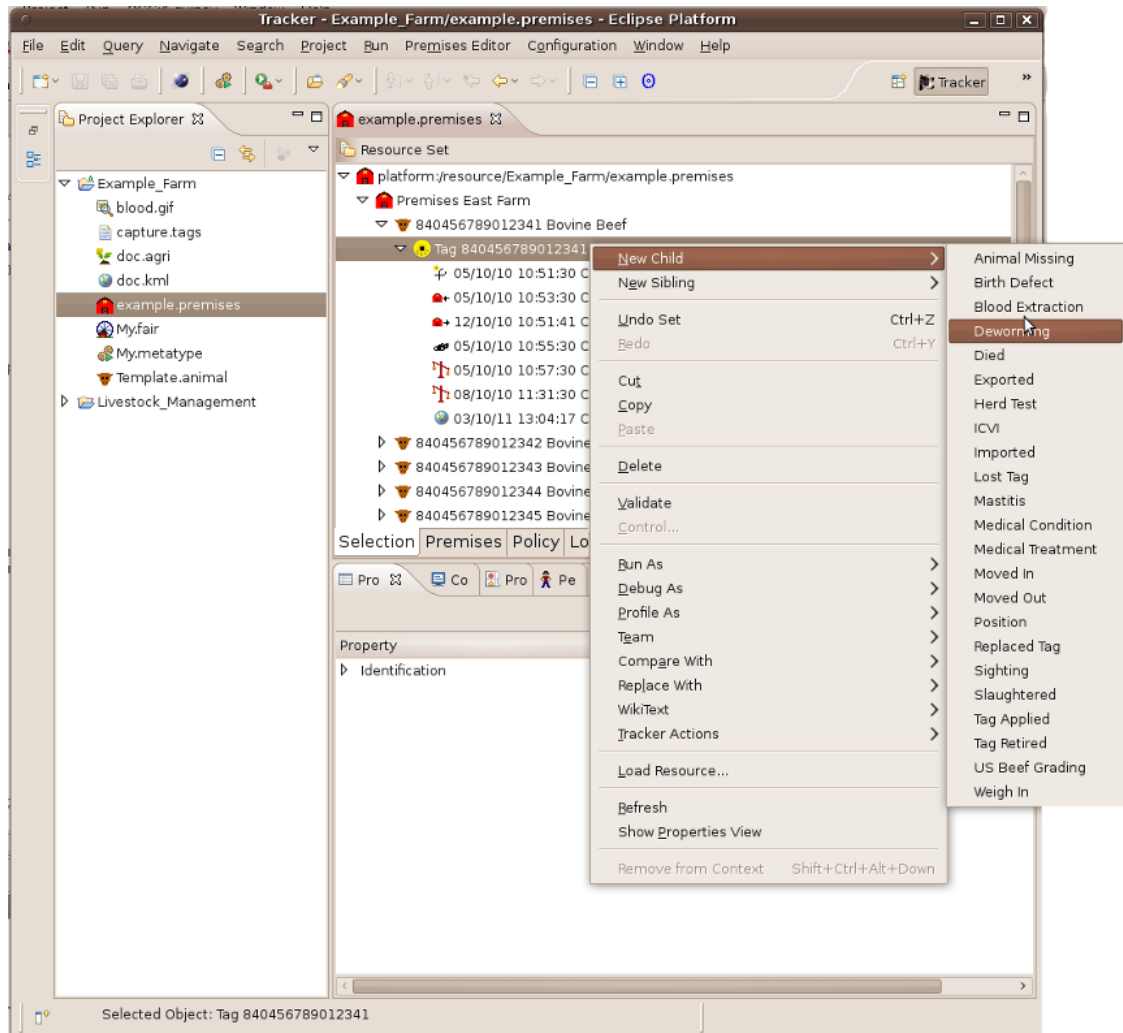
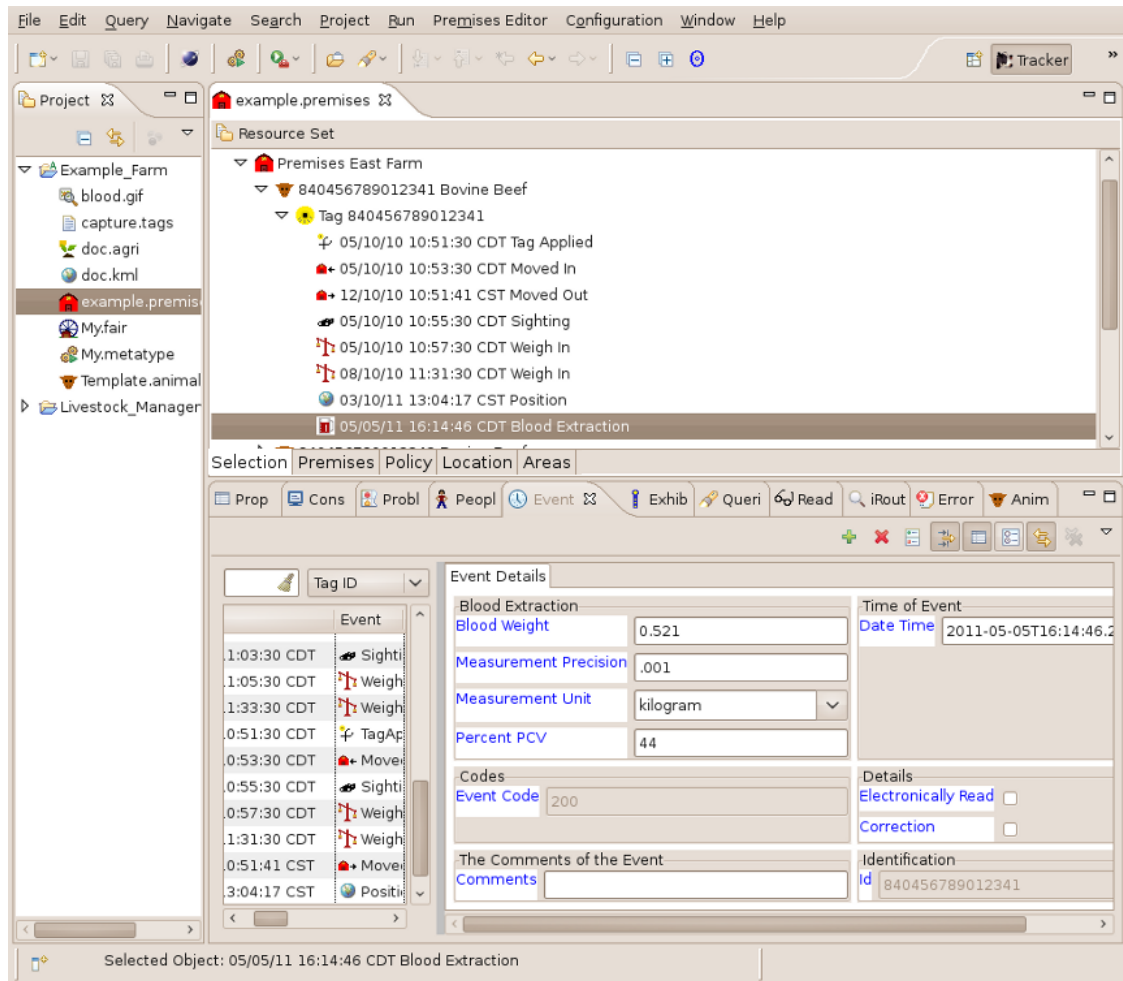
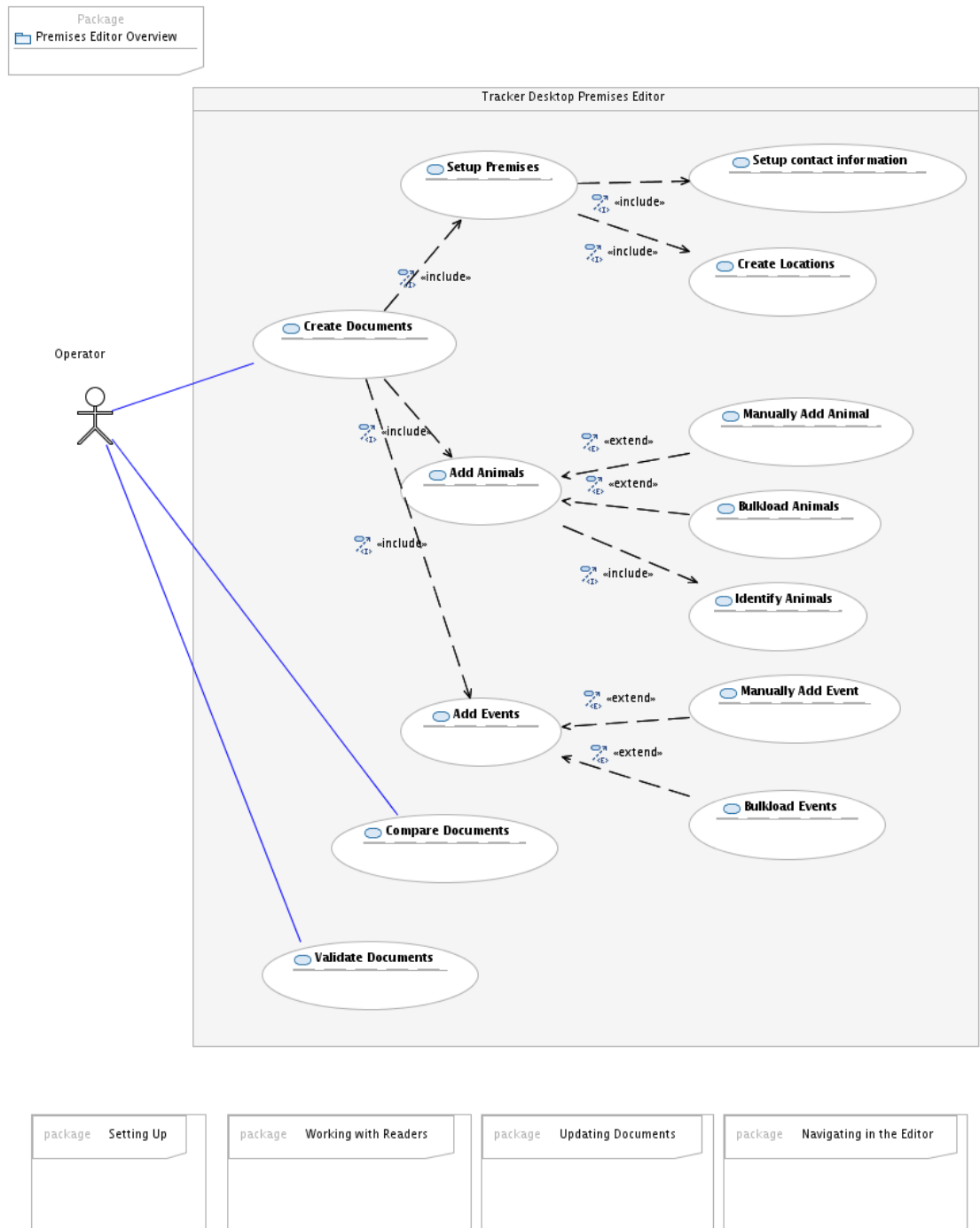
Figure 6.28. Adding an Event in the Premises Editor**Figure 6.29. Adding and Event in the Event View**

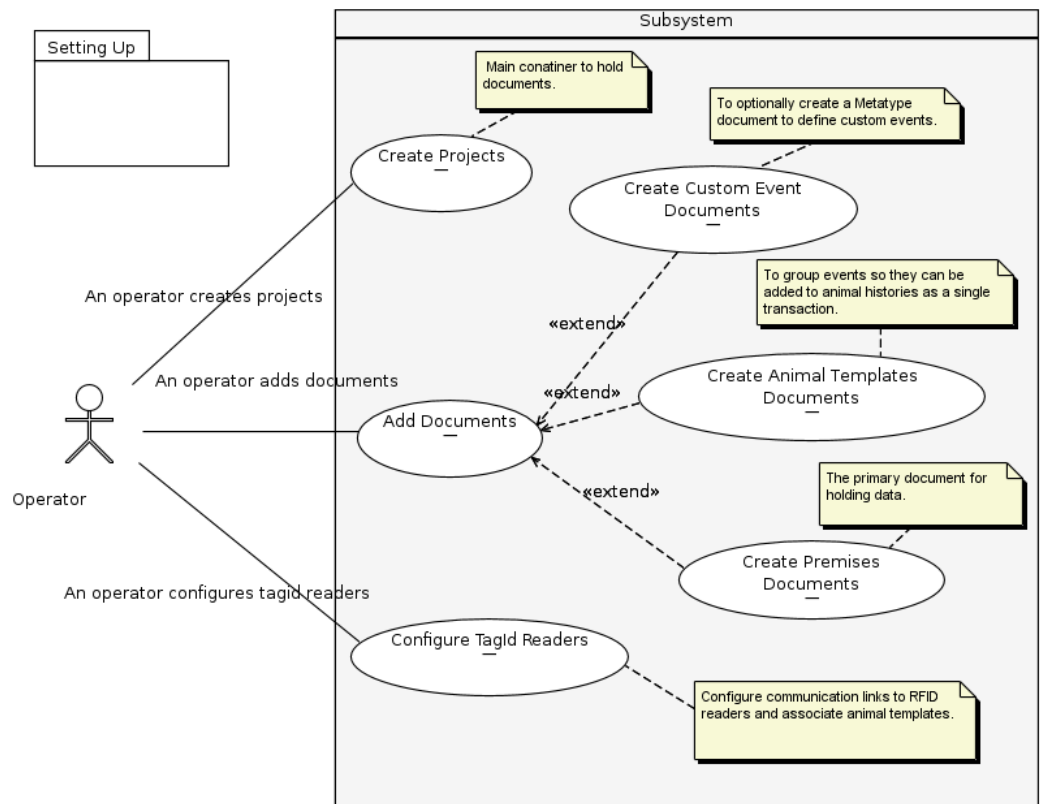
Figure 6.30. Modifying Attributes in the Events View

Premises Editor Use Cases

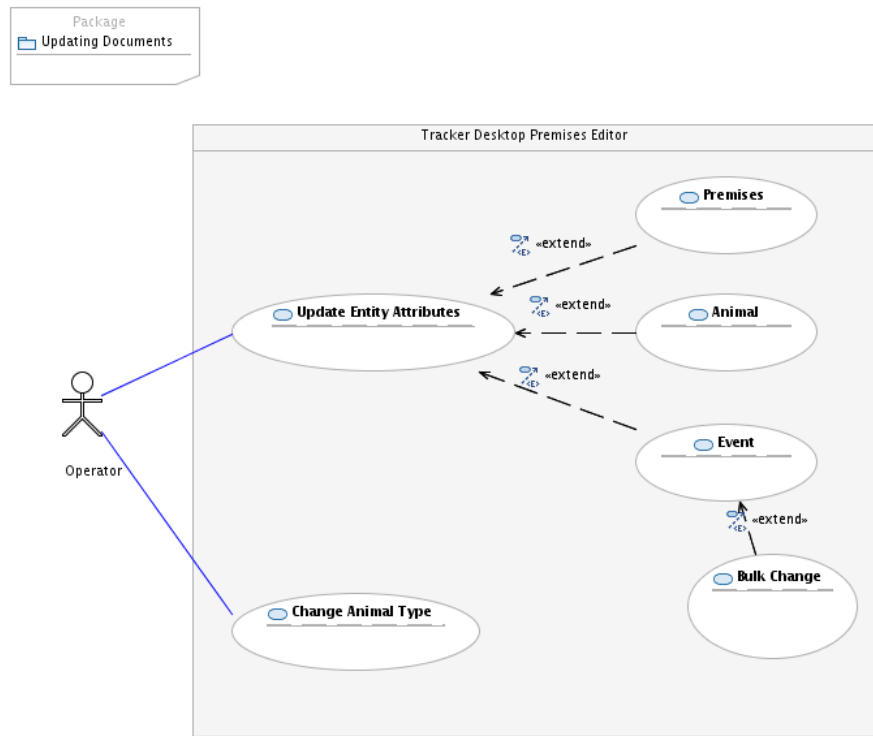
To best depict what the Premises Editor can do for Tracker users the following section presents a series of *Use-Case* diagrams. Figure 6.31, “Premises Editor Overview” shows core usage requirements for the Premises Editor.

Figure 6.31. Premises Editor Overview

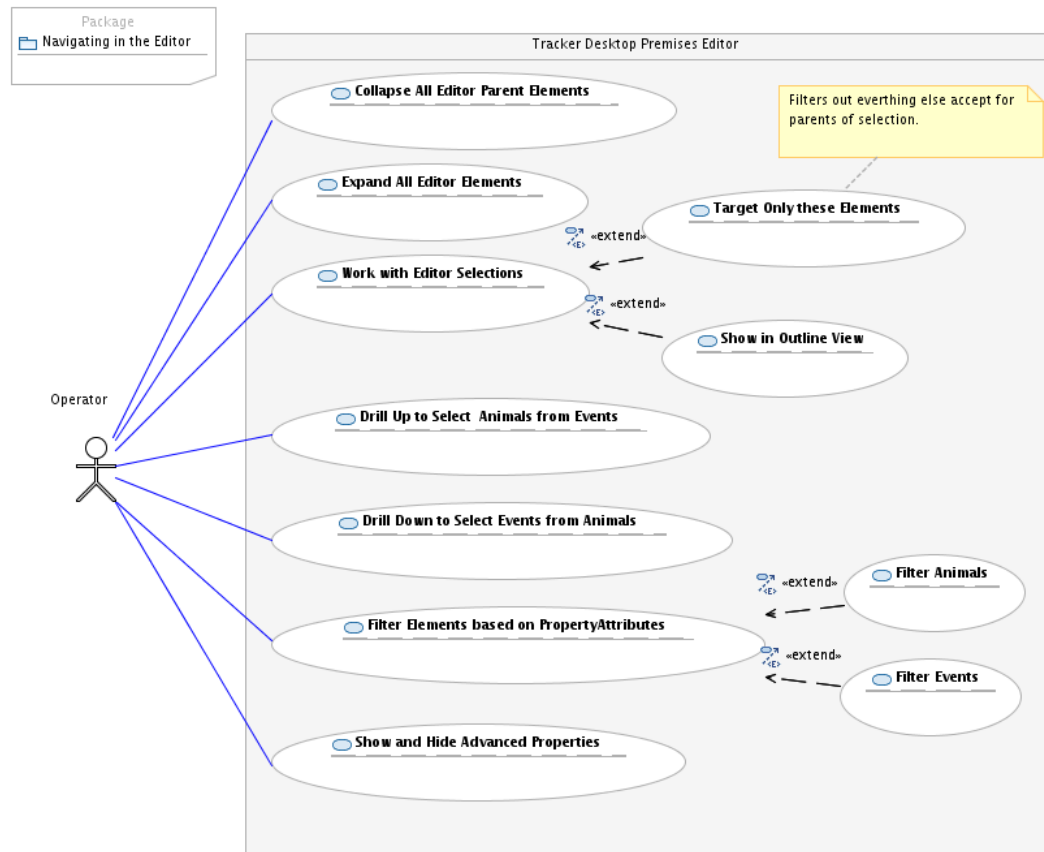
Once the Tracker BI Toolkit and the Premises Editor Feature are installed setup takes less than a minute. Administration use-cases are shown in Figure 6.32, “Setting Up the Premises Editor”.

Figure 6.32. Setting Up the Premises Editor

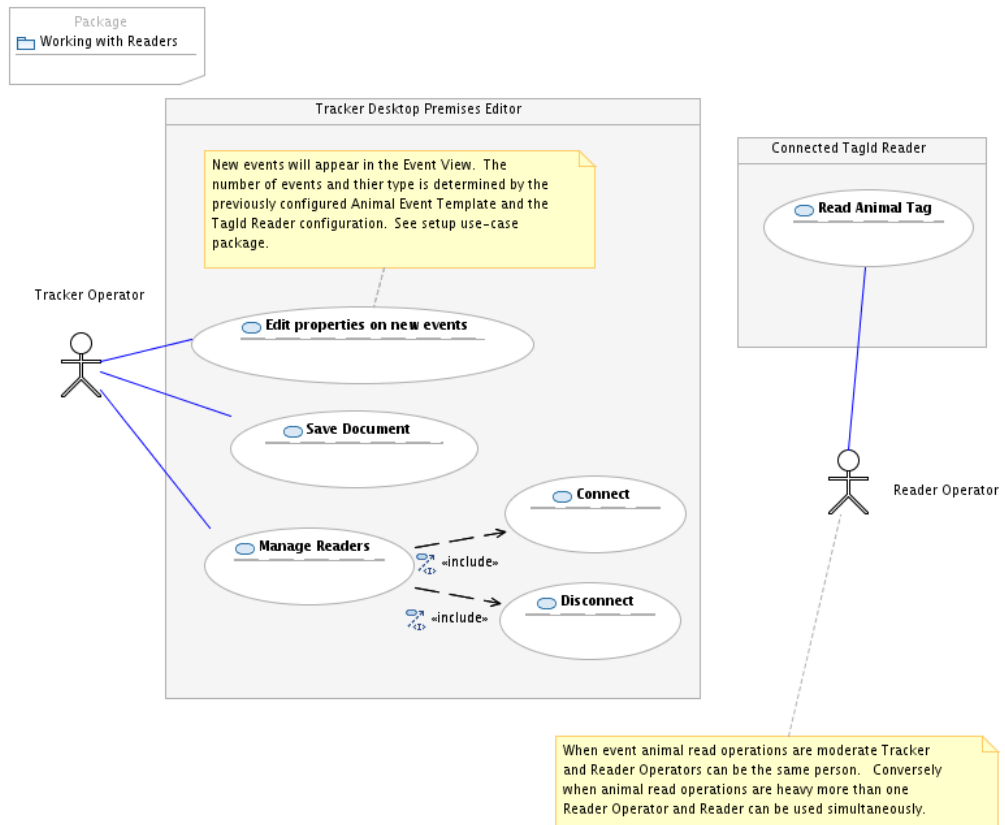
Updating document use-cases are shown in Figure 6.33, "Updating Documents".

Figure 6.33. Updating Documents

Tracker Operators may need to work with large Premises documents. Providing Tracker Operators with the tools to navigate around these large documents has been a major design goal of the Premises Editor. Navigation use-cases are shown in Figure 6.34, “Navigating in the Editor”.

Figure 6.34. Navigating in the Editor

As shown in previous use-cases Premises Documents can be created and updated by human operators Premises Editor. To improve efficiency new animals and events can also be added to Premises Documents by using external tag ID readers connected to the Tracker via data communications channels. This is shown in Figure 6.35, “Working with Connected Readers”.

Figure 6.35. Working with Connected Readers

Chapter 7. Metatype Model Editor

Overview

What is the Metatype Model Editor?

The Metatype Model Editor is a comprehensive open desktop solution for creating custom information *metadata* [<http://en.wikipedia.org/wiki/Metadata>] types. Metatypes formally specify the classes (ways) to describe an object. They are useful for providing a means to customize a structured process like a computer program.

Note

The Metatype Model Editor creates Metatype schema [<http://www.osgi.org/xmlns/metatype/v1.1.0/metatype.xsd>] compliant documents that can be used as input to *OSGi Metatype Service* [http://en.wikipedia.org/wiki/OSGi_Specification_Implementations#105:_Metatype_Service] and Configuration Administration Service [http://en.wikipedia.org/wiki/OSGi_Specification_Implementations#104:_Configuration_Admin_Service].

Project Goals

More often than not, new requirements are discovered by users after a program is deployed and operational. Providing the means to allow administrators and users to customize applications on site without waiting for a costly and time consuming new development cycle is the goal of the Metatype Model Editor project.

The Metatype Editor provide models, editors and utilities for the creation and management of documents that:

- Define object classes that specify one or more attributes of data
- Serve as the foundation for building new features that can be easily customized by users
- Conform to the OSGi Metatype schema
- Can be used as input to the OSGi Configuration Administration Service

Functional Overview

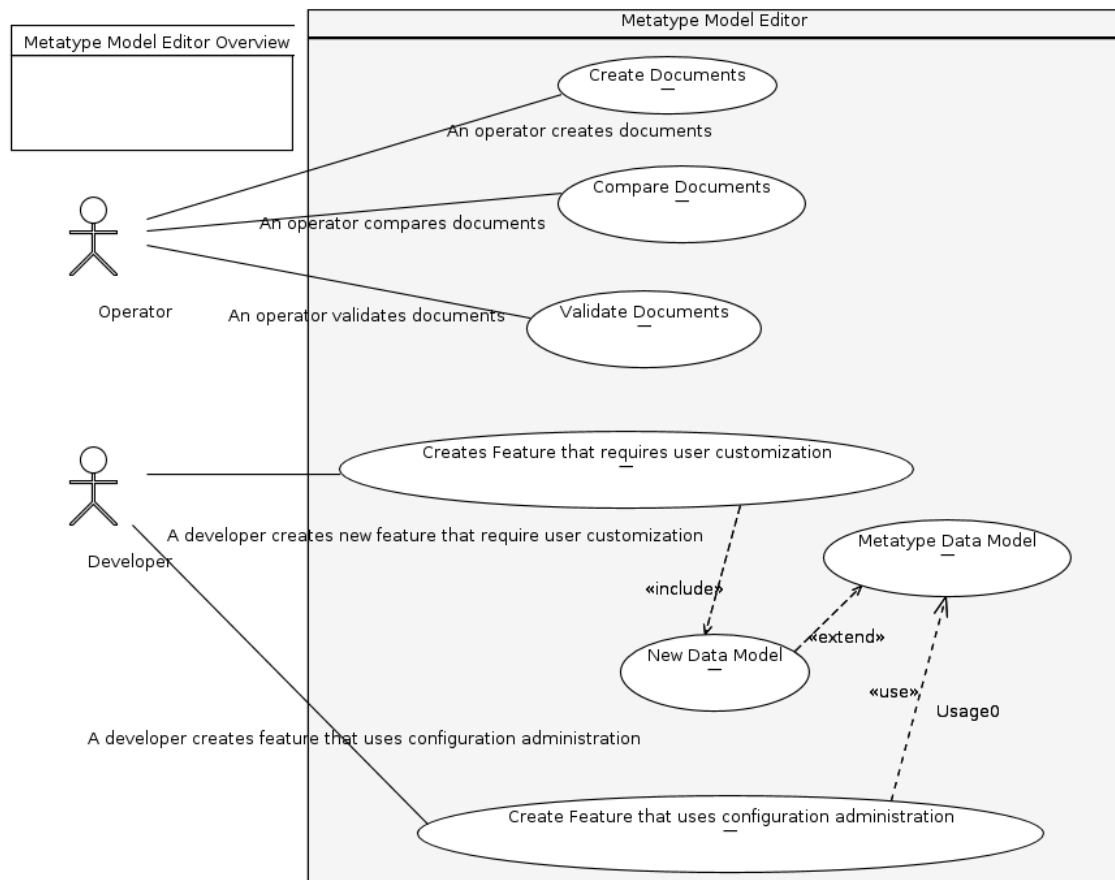
The Metatype Model Editor Feature is an implementation of the OSGi Metatype data model and an editor to create metatype model document resources. Metatype document resources can be utilized by other features enabling feature specific customization by end users.

The OSGi Metatype is based on the idea of a user created model document that specifies how data in an target application is named, entered and structured. In other words, Metatype model documents provide a means to classify sets of user defined attribute definitions that can act as input definitions to customize the behavior of applications and/or application features.

Metatype Data Model and Model Editor Use Cases

The following figure presents a *Use-Case* diagram showing how the Metatype Data Model can be extended by developers to create features with customizable functionality and how the Metatype Model Editor is used by users to author documents that define application or feature specific customizations. Figure 7.1, “Metatype Data Model and Metatype Editor Use Cases” shows core use cases for the Metatype Data Model and Model Editor.

Figure 7.1. Metatype Data Model and Metatype Editor Use Cases



Metatype Documents

The Metatype Editor enables users to create, transform and edit metatype documents to define new functionality.

Note

Metatype Documents are only needed when they are used by features to customize functionality.

Metatype document files utilize a `.metatype` file extension to distinguish them from other types of files.

Metatype Data Model

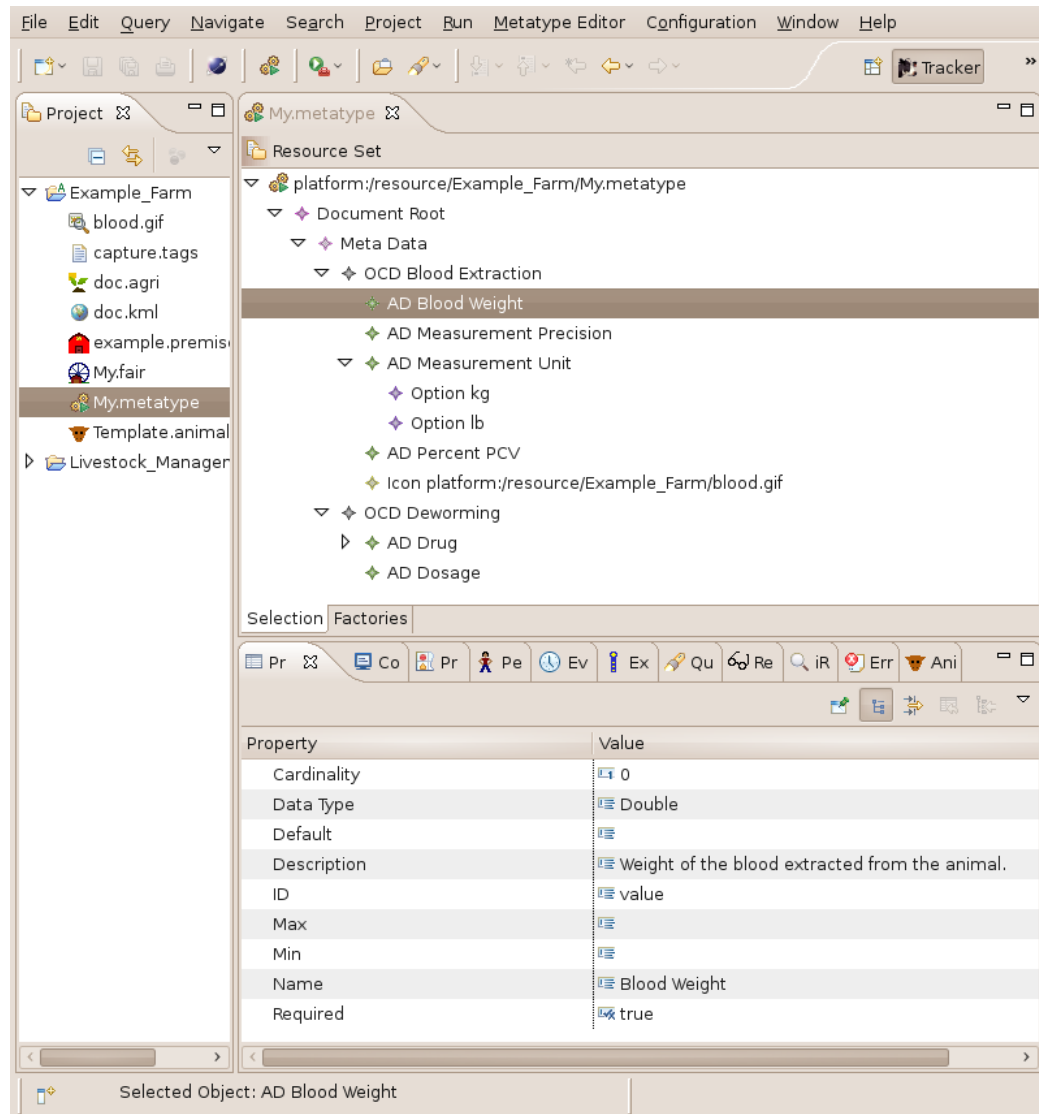
Because all user activity centers around the creation and manipulation of *Metatype Documents*, understanding the structure (called the *data model*) of Metatype documents is the easiest way to learn to author Metatype model documents.

Data Model Core Elements

All Metatype documents conform to a simple structure called the *Metatype Data Model*. This structure is based on the idea of groupings of one or more data attribute definitions.

In the Metatype Data Model, a single root element called *Metadata* acts as a container for one or more Object Class Definitions called *OCD* elements that are data type containers or parents for Attribute Definitions called *AD* elements. These *AD* elements are the fundamental building blocks for defining the names and types of data associated with a *customization*.

For example, in the problem domain for managing information that is associated with livestock, an application may require that users add custom functionality like a user interface, for recording actions done to treat livestock illness. The Figure 7.2, “Metatype Model Document” screenshot shows the editing of two *OCD* elements in a Metatype model document. When this document is used as configuration input for the application, the user could be presented with two new user interfaces that are added a data input screen. The first for recording a *Blood Extraction* treatment and a second for recording a *Deworming* treatment.

Figure 7.2. Metatype Model Document

The following table explains the Element Relationships of the Metatype Model.

Table 7.1. Core Elements of the Metatype Model

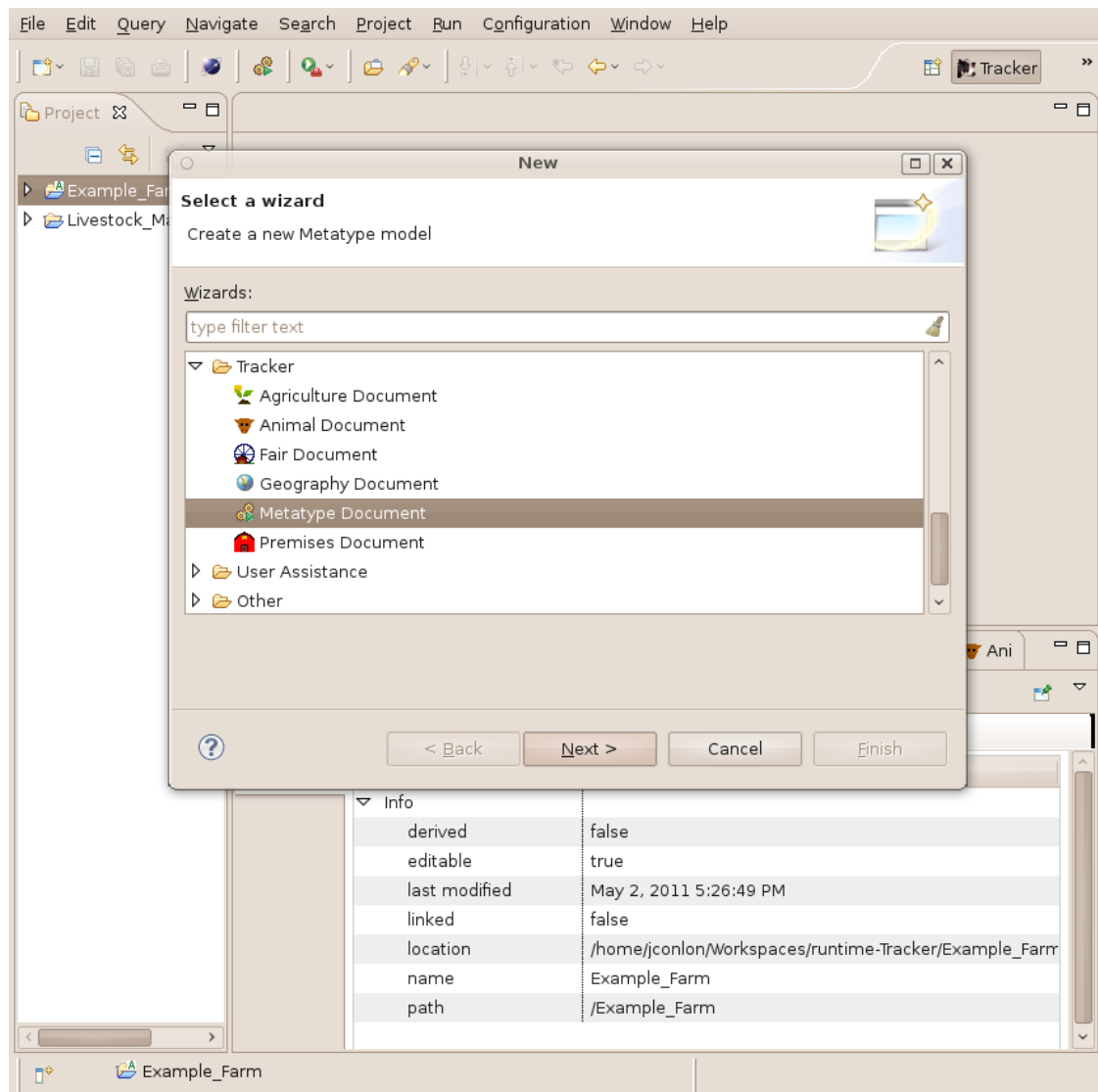
| Element Name | Relationship to other Elements |
|--------------------------------|-------------------------------------------------------------------------------------------------------------------|
| Meta Data | There is one Meta Data element and acts as the root element in a document |
| OCD - Object class definitions | Meta Data can have one or many OCD elements |
| id, description, name | OCD element has an id, description and name attributes |
| AD | OCD can have one or many AD element |
| AD attributes | The properties view of the Figure 7.2, “Metatype Model Document” screenshot shows the attributes of an AD element |

Editing Metatype Documents

Creating a new Document

To create a new metatype document. Go to the File menu and launch a *New Wizard* by selecting the menu items: `File | New | Other`. Figure 7.3, “New Wizard for Creating a New Metatype Document”.

Figure 7.3. New Wizard for Creating a New Metatype Document



Follow the instructions to choose a project, folder and a name for the document and press **Finish** to create the document.

Note

For more details about authoring Metatype Model documents refer to the user guide of the feature using Metatype documents for customization.

Chapter 8. Geography Editor

Overview

What is the Geography Editor?

The Geography Editor is an open source desktop solution for authoring, managing and integrating state of the art Geographic Information System GIS technologies based on the *OpenGIS* [<http://www.opengeospatial.org/>] Key Hole Markup Language KML.

Project Goals

The KML is an industry standard GIS markup language for creating documents for presenting geography information in 2 dimensional maps and 3 dimensional *Geobrowsers* [http://en.wikipedia.org/wiki/Digital_Earth#Geobrowsers]. The Eclipse Modeling Framework *EMF* [<http://www.eclipse.org/modeling/emf/>] is an Eclipse-based modeling framework and code generation facility for building tools and other applications based on a structured data model.

The primary goal of the Geography Editor project is to build an EMF data model and editor for KML to provide the foundation for easily integrating location information into EMF tools, applications, and other EMF based data models.

Functional Overview

KML documents can be created from any of the free or commercially available XML based editors or from utilities associated with many of the KML based Geobrowser applications. While the Geography Editor Feature provides an alternative Editor for authoring KML documents, the Geography Editor Feature's primary function is to enable the integration of all KML documents, no matter how they were originally created, with the EMF application framework in order to leverage existing and future EMF based tools, utilities and data models.

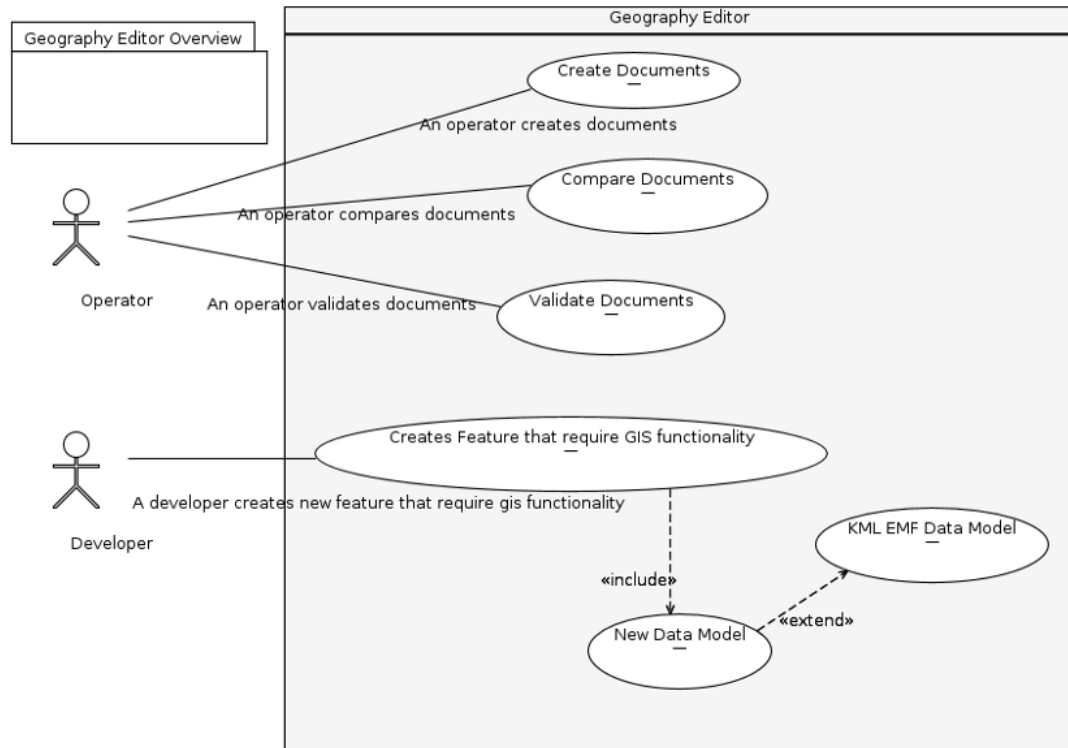
Note

While the Geographic Editor Feature does provide a KML editor, it's fundamental purpose is to act as a base technology by offering a KML/EMF data model, that can be easily extended by other features that require GIS data integration.

Geography Editor Use Cases

The following figure presents a *Use-Case* diagram showing how the Geography Editor can be used by operators and how it can be extended by developers to create new functionality. Figure 8.1, "Geography Editor Use Cases" shows core usage requirements for the Geography Editor.

Figure 8.1. Geography Editor Use Cases



Geography Documents

The Geography Editor enables users to create, transform and edit KML documents. See *KML Introduction* [<http://code.google.com/apis/kml/documentation/>] for basic information about creating and editing KML documents.

Note

Geography document files utilize a .kml file extension to distinguish them from other types of files.

Geography Data Model

All operator activity centers around the creation, manipulation and visualization of *Geography Documents*. To understand the structure (called the *data model*) see the *KML Reference* [<http://code.google.com/apis/kml/documentation/kmlreference.html>] for more information.

Note

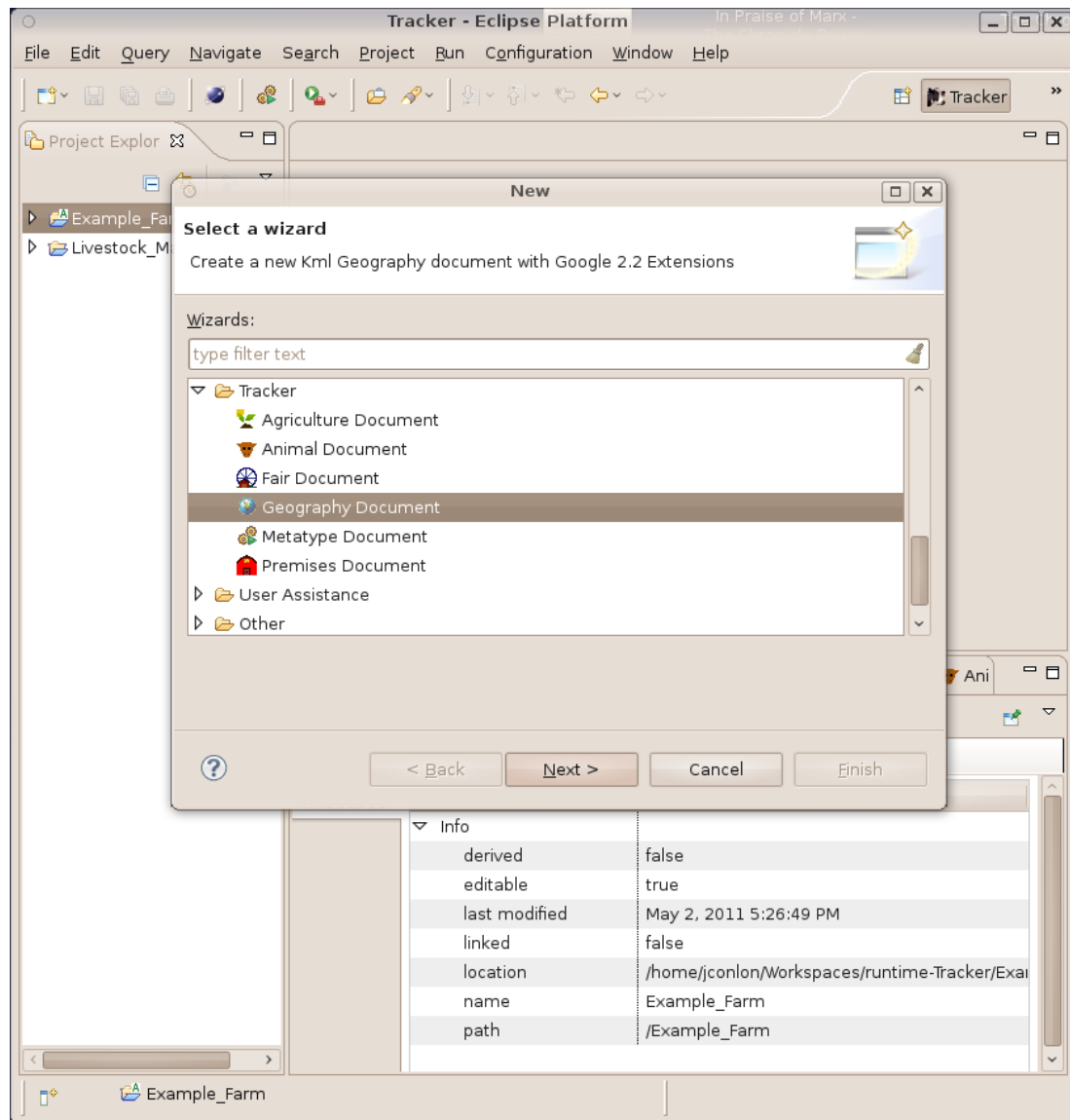
The Geography Data Model conforms to *Google's Extensions to KML*. [<http://code.google.com/apis/kml/documentation/kmlreference.html#kmlextensions>]

Editing Geography Documents

Creating a new Document

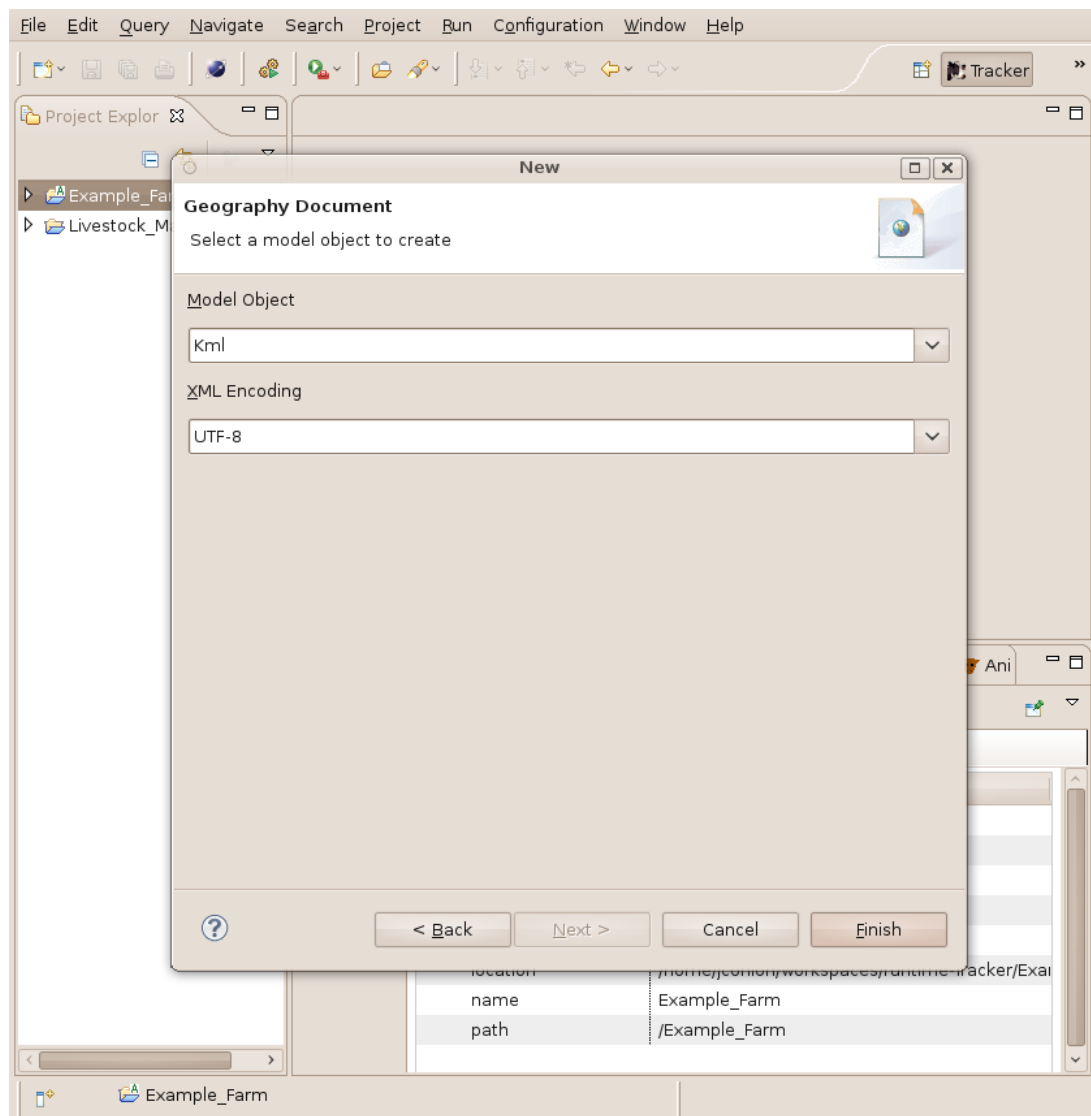
To create a new Geography document. Go to the File menu and launch a *New Wizard* by selecting the menu items:File | New | Other. Figure 8.2, “New Wizard for Creating a New Geography Document”.

Figure 8.2. New Wizard for Creating a New Geography Document



Follow the instructions to choose a project, folder and a name for the document. Select one of the KML elements in the list for the root of the document and press **Finish** to create the document.

Figure 8.3. New Geography Document Wizard Prompt



Chapter 9. Agriculture Editor and Agriculture Location Service

Overview

What is the Agriculture Editor and Location Service?

The Agriculture Editor is an editor for specifying business associations of livestock producers and infrastructure providers.

The Agriculture Location Service is a comprehensive desktop based implementation of the Tracker Location Service ¹. The Agriculture Location Service implementation utilizes local desktop resident Agriculture document files to create a geolocation map for resolving geolocation requests.

The Agriculture Editor and Location Service can be used for documenting, resolving and visualizing livestock herd distributions; identifying livestock business associations; and determining individual animal locations, and animal movements within and between premises locations.

Project Goals

As livestock markets continue to adapt to customer requirements, more livestock markets will require animal source information and the functionality to do trace-back and trace-forward between business associates. Yet even for some producers not facing this inter-premises requirement, just resolving animal locations, within a premises, is a must have requirement.

Making livestock location information easy to utilize and easy to electronically inter-exchange while maintaining data privacy within business associations of livestock producers is the goal of the Agriculture Editor and Location Service project.

The Agriculture Editor and Location Service provide models, editors and services for the creation and management of Agriculture documents that:

- Specify associated groups of business related livestock premises ² within geographic regions
- Identify agriculture documents that will contribute location data to the Location Service.
- Locate Animals that have moved to geographical sub-locations within a premises
- Locate Animals that have moved to another premises within associated groups of business related livestock premises
- Identify a list of sub location areas within premises

¹ The Tracker Location Service is a global Tracker application service for resolving the real-world geographic location of an object.

Given the name or identification (ID) of an object, the Location Service can resolve geographic location names, addresses, and sub areas of the object. Given the geographic coordinates of an object the service can resolve the name of the containing location.

The Tracker Location Service aggregates all implementations to provide a single service application interface to enrich Tracker desktop editors and utilities.

²It is the optional 'Location' element and child 'Area' elements in the Tracker 'Premises' document that specifies the geographic information that will be used to create the Agriculture Location Service mapping.

- Assign animal positions based on visual sighting records or data inputs from industry standard GPS devices
- Will serve as the foundation for a full featured distributed livestock Traceability service

Functional Overview

The Agriculture Editor creates documents that assign Livestock premises to Associations.

Note

The Agriculture Editor is dependent on documents created by the Tracker Premises Editor. The Agriculture element `Association` specifies one or more references to `Premises` elements in *.premises documents

The Agriculture Location Service provides a geolocation resolution service that is based on the information contained in specially designated Agriculture documents.

The Agriculture Location Service implementation adds geographical visualization, location resolution and geographic analytical utilities to the Tracker Desktop Business Intelligence Toolkit:

- Desktop editors, views, and utilities for modeling agriculture associations allowing livestock producers auditors, and processors to geographically -
 - manage internal, inter-association and intra-association animal movements and information flows
 - create marketing and/or auditing reports for regional and *local* based farm products
- Easily view the location of animals within the local premises.
 - Resolution of locations based on animal Position events that were input from GPS devices associated with RFID readers. Where Position coordinates determine the location names based on best fit (smallest area) of list of geographically specified locations
 - Visual sighting of animals with manual assignment of location names selected from list of geographically specified locations
 - Most current Sighting event or Position event determines location
- Easily view the destination of animals moved to other business partners

Agriculture Documents

The Agriculture Editor enables users to create and edit documents that specify associations Livestock Premises.

Note

Because Agriculture Documents elements references elements in one or more Premises, the Premises documents need to be created before the user creates an Agriculture document.

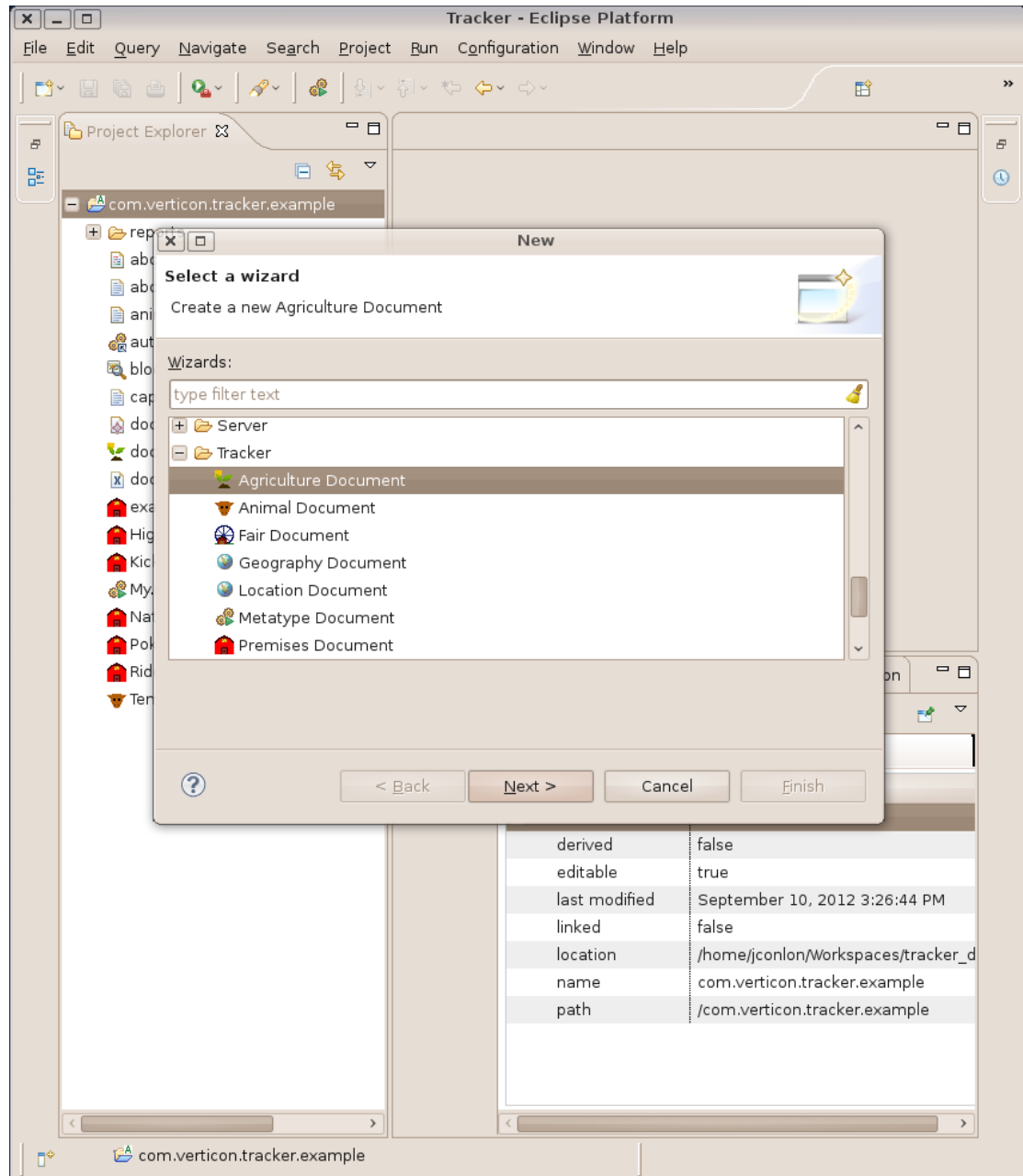
Agriculture document files utilize a `.agri` file extension to distinguish them from other types of files.

Editing Agriculture Documents

Creating a new Document

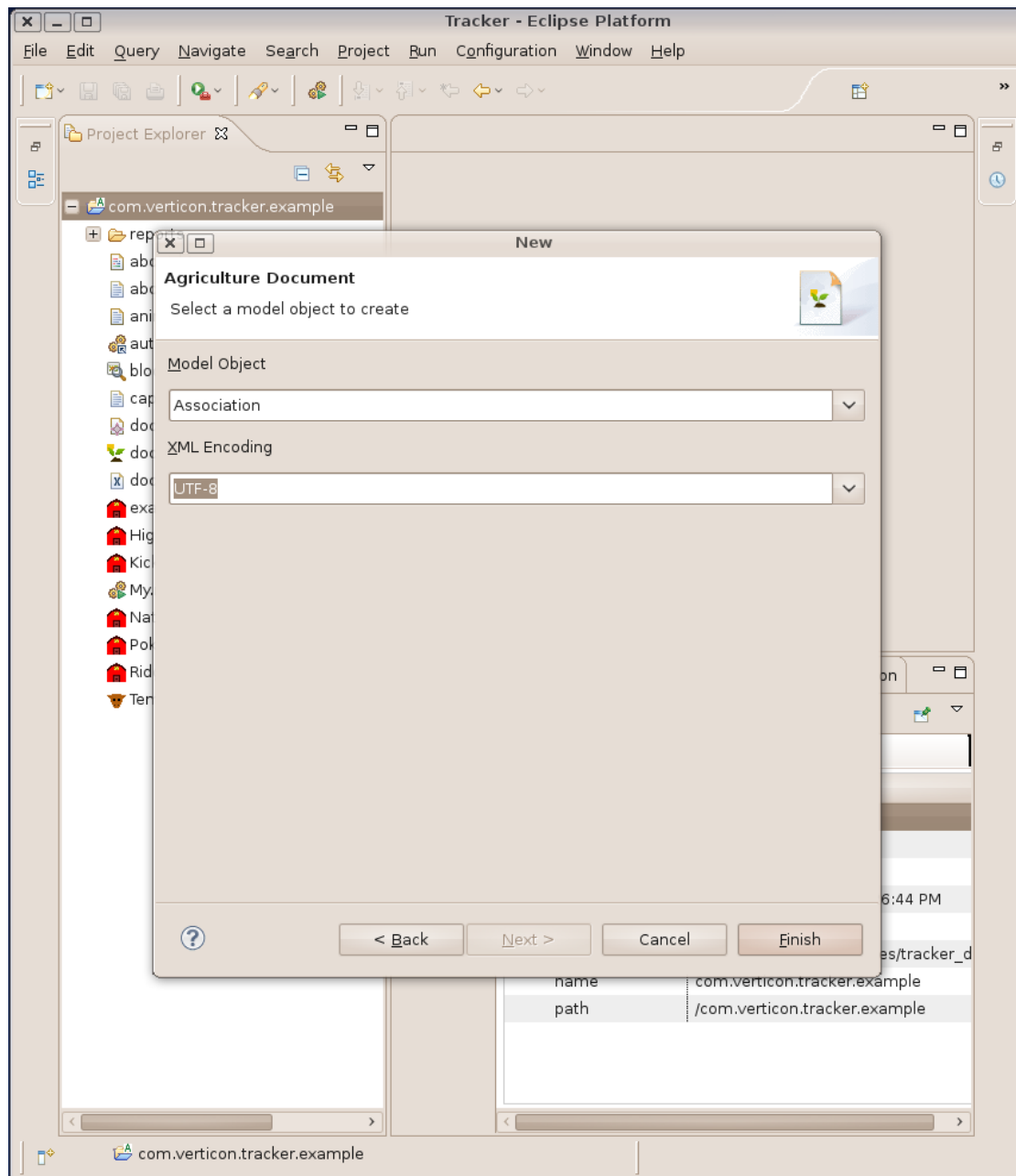
To create a new agriculture document. Go to the File menu and launch a *New Wizard* by selecting the menu items: File | New | Other.

Figure 9.1. New Wizard for Creating a New Agriculture Document



Follow the instructions to choose a project, folder and a name for the document. Select Association as the model object and press **Finish** to create the document.

Figure 9.2. New Agriculture Document Wizard (Model Object Prompt)



Loading Additional Resources into the Document

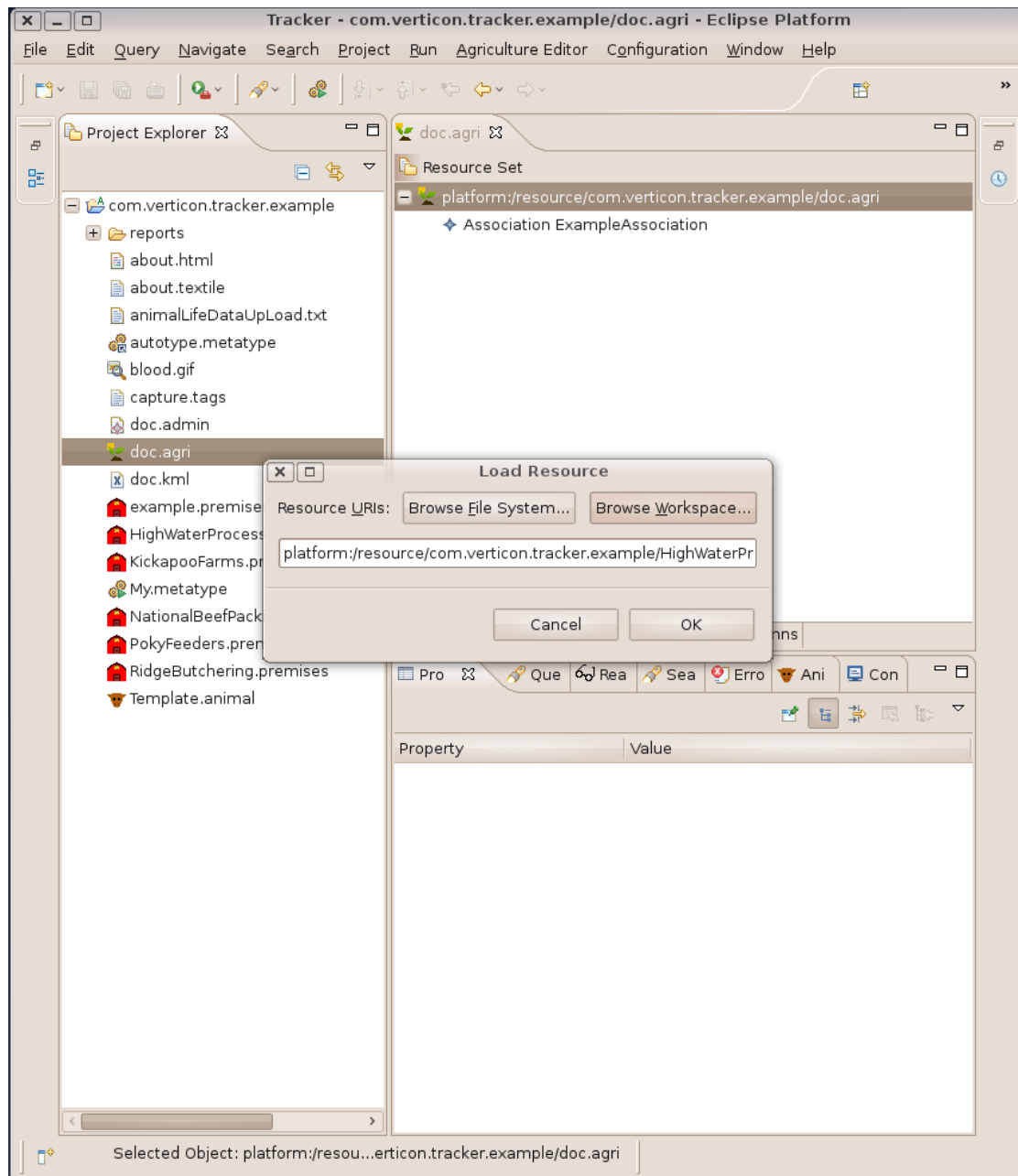
Note

Since the Agriculture document references Premises elements, one or more of these Premises documents must exist as resources in the workspace.

To enable the Agriculture Editor to create references to Premises the Premises documents must be loaded as resources in the Agriculture document. When all the needed Premises documents are loaded as resources to the Agriculture document, these Premises elements can then be referenced as values of the Association Livestock attribute.

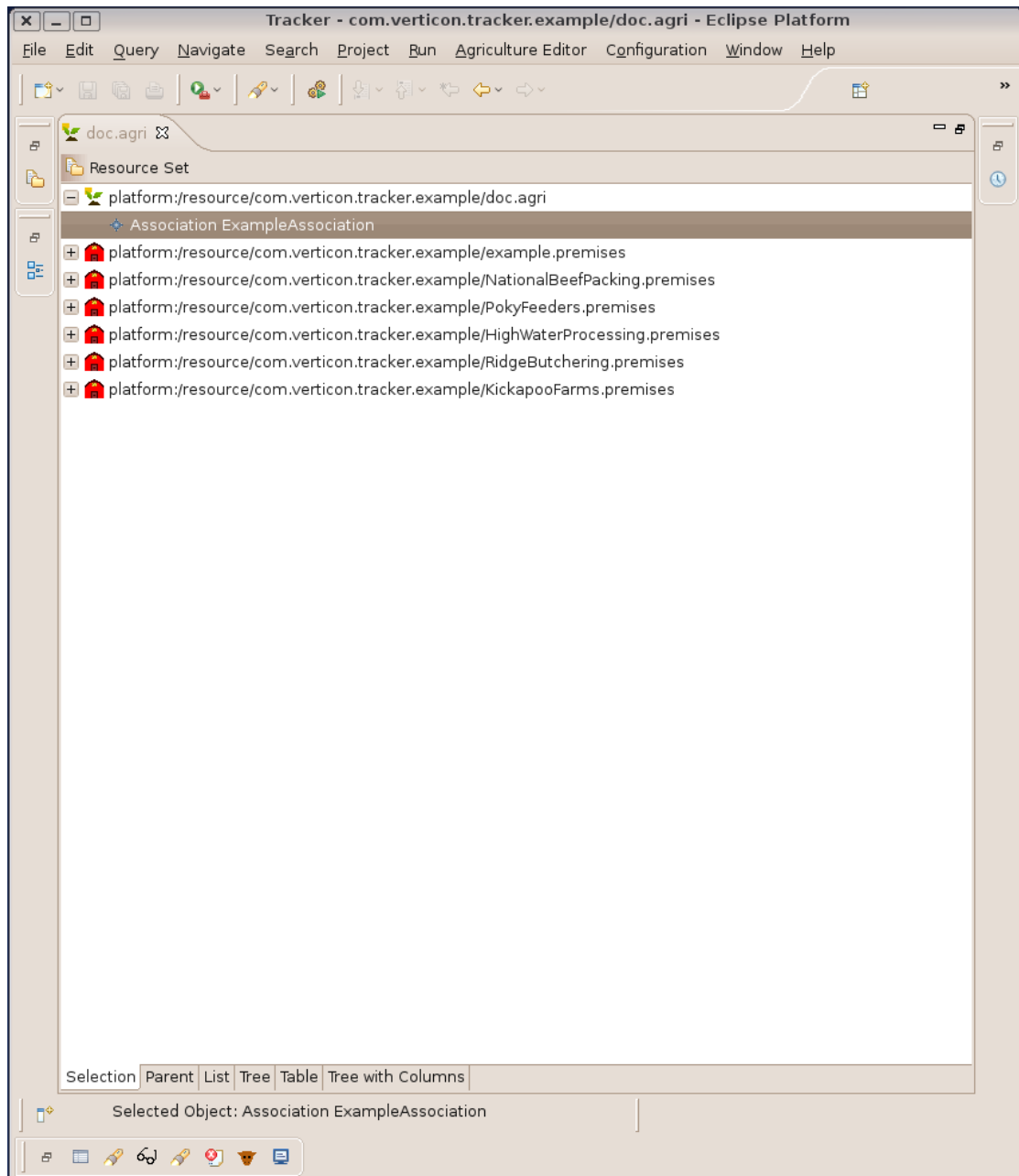
To load a resource place the mouse cursor on the root element in the document, right click, and from the pull down menu choose Load Resource...

Figure 9.3. Load Resource Dialog



From the dialog choose the KML or the Premises document that you wish to load. Repeat the loading process to load all the resources that will be referenced by the Agriculture document.

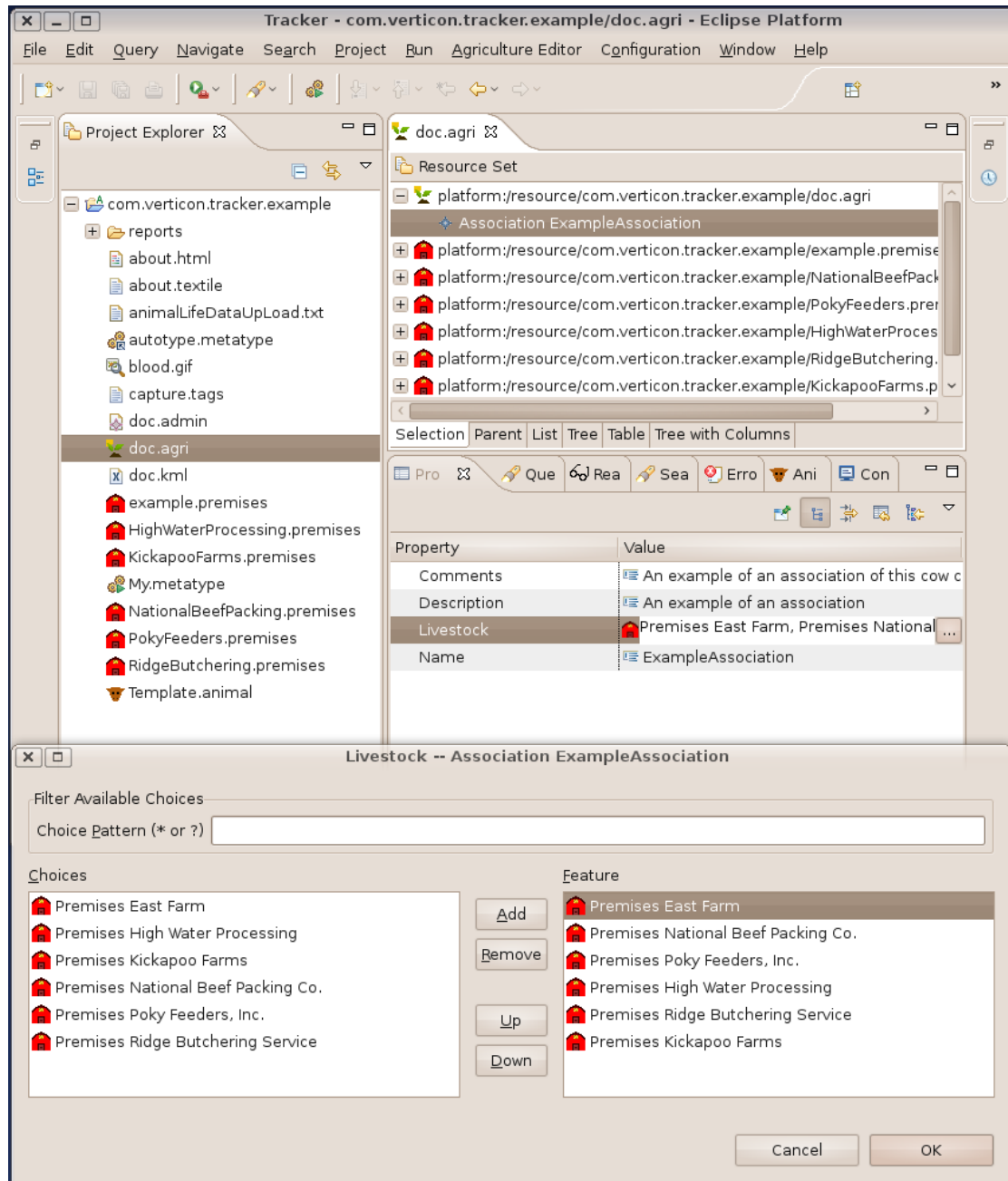
Figure 9.4. Agriculture Project with Loaded Resources



Completing an Agriculture Document

When premises documents are loaded as resources into Agriculture documents, the editor screens associated with the `Livestock` attribute of the `Association` element will be populated by all `Premises` elements from all the loaded `Premises` resources.

Figure 9.5. Picklist of Premises



Agriculture Data Model

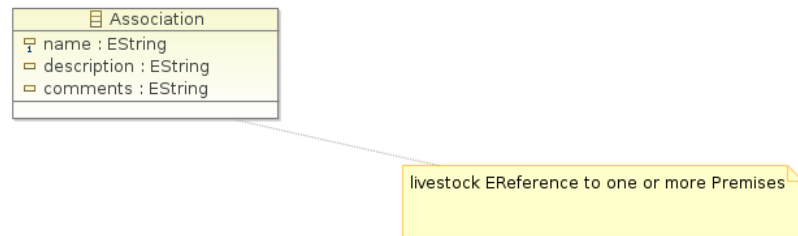
Because all user activity centers around the creation and manipulation of Agriculture Documents *Association*, understanding the structure (called the *data model*) of Agriculture documents is the easiest way to learn to how to create documents with the Agriculture Editor.

Data Model Core Elements

All Agriculture documents conform to a very simple structure called the *Agriculture Data Model*. In the Agriculture Data Model, there is only a single element called an *Association*. The Association has three simple text based attributes, `name`, `description` and `comments`; and one complex attribute `livestock`. The `livestock` attribute is a multivalued attribute that references one or more Premises elements.

Core elements of the Agriculture Data Model are shown in Figure 9.6, “Agriculture Data Model Elements”.

Figure 9.6. Agriculture Data Model Elements



- The only element of the Agriculture Model, Association is depicted as a rectangle in the figure.
- The notation within the Association element are its attributes.
- The notation outside of the Association element represents the `livestock` attribute's references to external Premises document elements.

The following table offers a text explanation of the Association elements relationships.

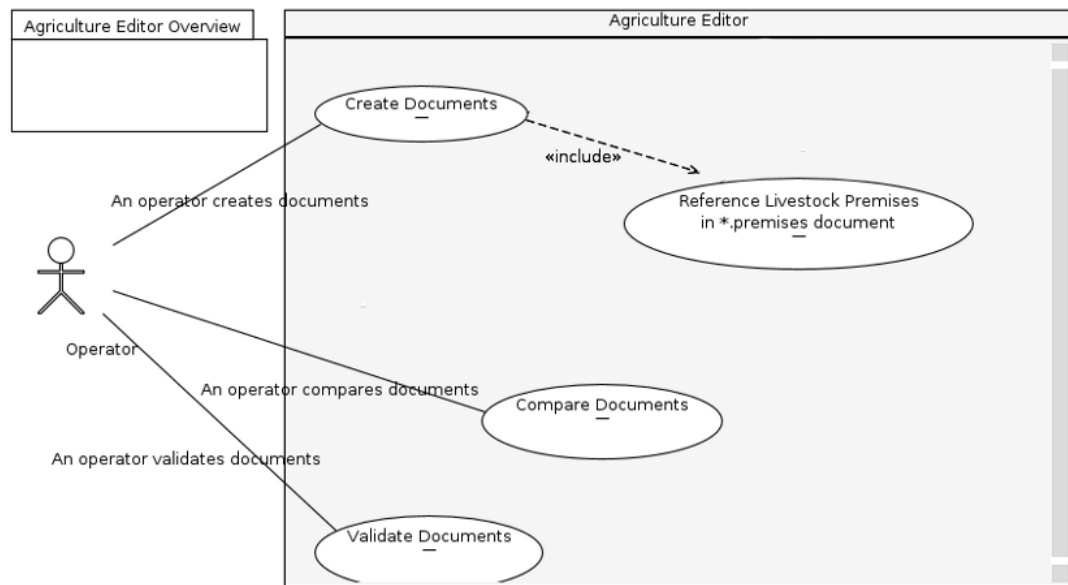
Table 9.1. Core Elements of the Agriculture Model

| Element Name | Relationship to other Elements |
|--------------|--------------------------------------------------------------------------|
| Association | This is the only element in a document. |
| name | Association has a name attribute. |
| description | Association has a description attribute. |
| livestock | Association has a multivalued livestock attribute. |
| Premises | Multiple Premises elements can be referenced by the livestock attribute. |

Agriculture Editor Use Cases

To best depict what the Agriculture Editor can do for Tracker users, the following figure presents a series of *Use-Case* diagrams. Figure 9.7, “Agriculture Editor Overview” shows core usage requirements for the Agriculture Editor.

Figure 9.7. Agriculture Editor Overview



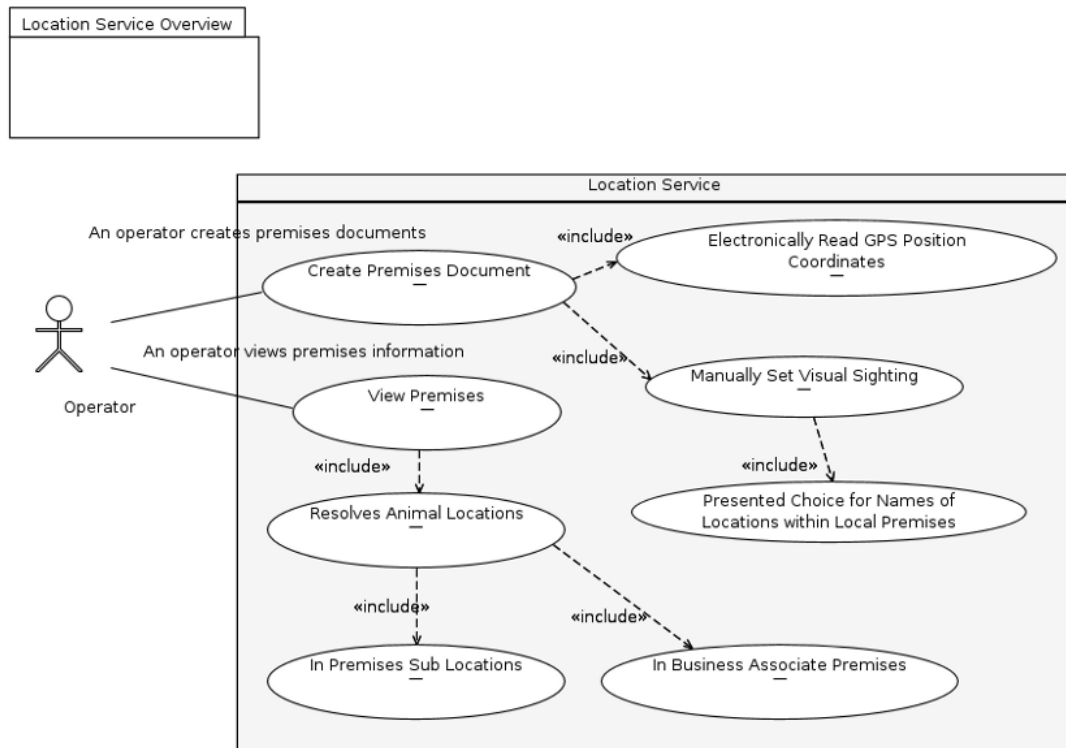
Location Service Use Cases

The Location Service is not directly accessible to the user, but operates in the background to provide geographical location resolution services to the Premises Editor and Premises document related tools. As a background service it resolves location specific elements and attributes within Premises documents, Premises utilities, and Business Intelligence reporting tools.

To provide location resolution, the Location Service derives it's data from one or more of Agriculture documents created by the operator with the Agriculture Editor.

The Location service provides the following *Use-Cases* to Tracker users. Figure 9.8, "Location Service Overview" shows core usage requirements for the Location Service.

Figure 9.8. Location Service Overview



Configuring the Location Service

Agriculture Documents as Input to the Location Service

The full benefit of declaring premises and geographical locations is achieved when Agriculture documents are defined as input to the Location Service.

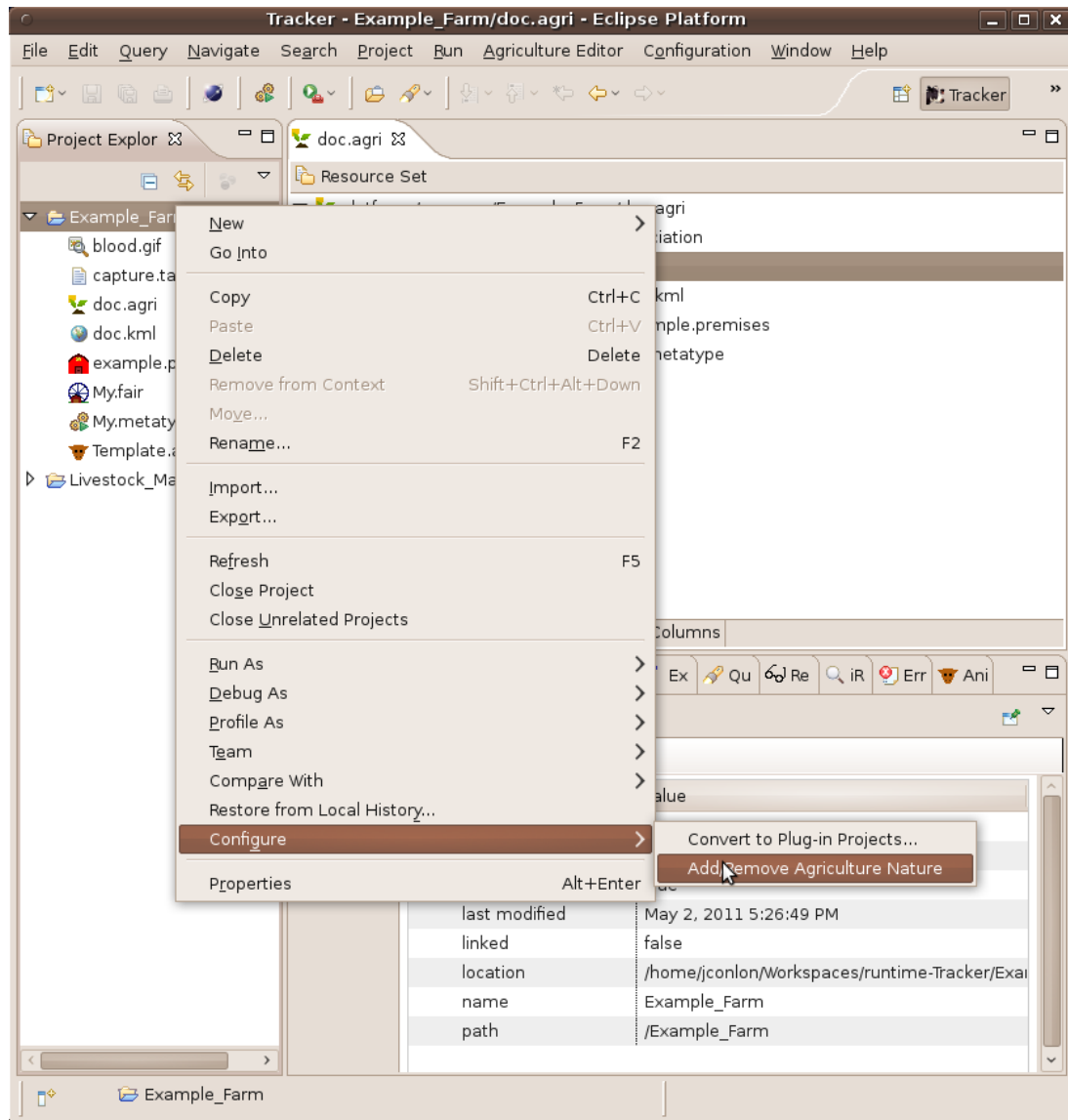
In order to assign an agriculture document to the Location Service. The following requirements must be met:

- The agriculture document must be a valid Agriculture documents
- The agriculture document must be named: `doc.agri`
- The agriculture document must be located in the root folder of a project
- The agriculture document must be in a project that has been configured with an Agriculture Nature

Configuring a Project with an Agriculture Nature

To configure a project with an Agriculture Nature place the mouse cursor on the Project in the Project Explorer right click, and from the pull down menu choose Add/Remove Agriculture Nature.

Figure 9.9. Configuring Project with an Agriculture Nature



Note

A project that has been configured as an Agriculture Nature will be designated with a *small capital A* on the upper right of the project folder icon.

Using the Location Service

Once a premises has been defined in the `doc.agri`, placed in the root directory of a project, and the project configured with an Agriculture Nature, the Location Service will provide resolution services for all of Premises documents in the workspace:

- Resolving the Premises element's derived `address` attribute to the address in the referenced Premises Location element.

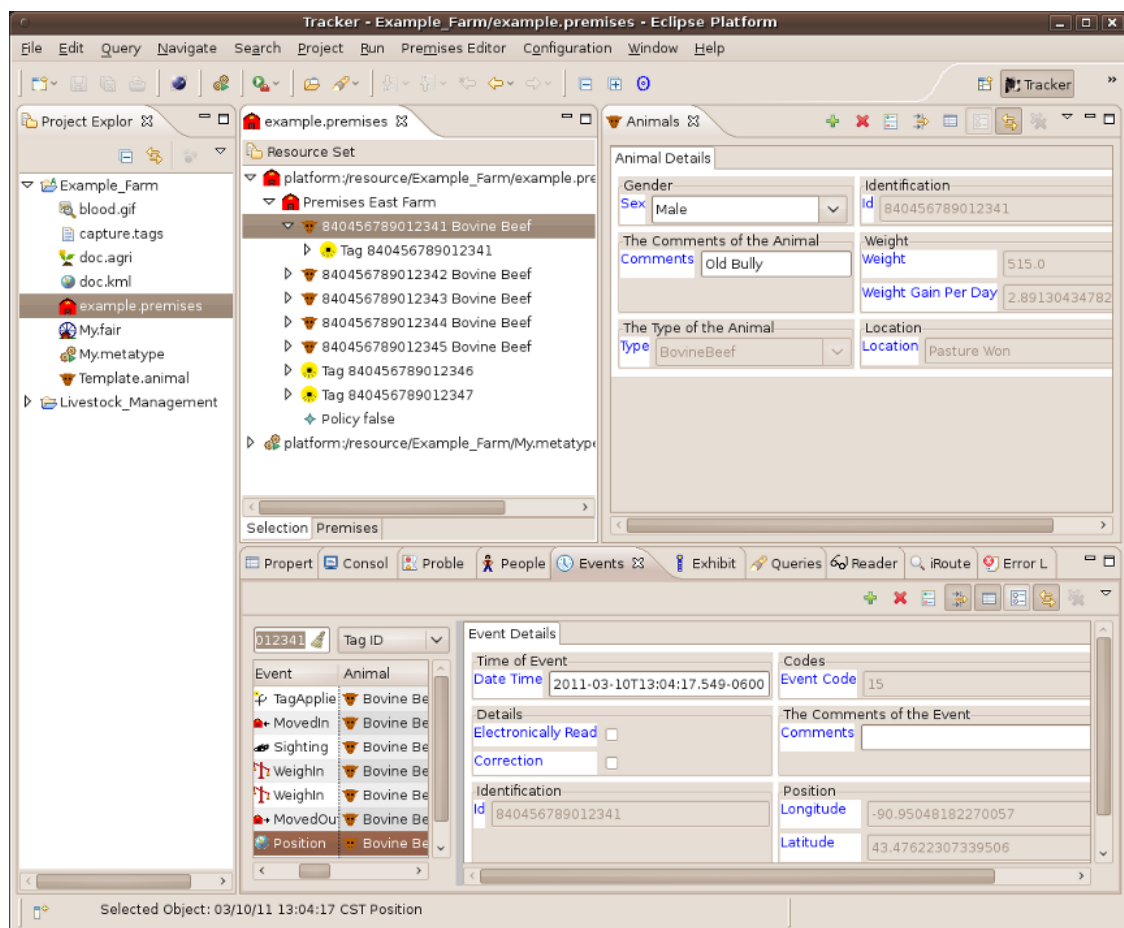
- Resolving the Animal element's' derived location attribute when the last movement event in the Animal event history is a:
 - Position event. Resolution will be the sub location within the Location Area boundaries that contains the point specified by the coordinate attribute of the Position event. See Figure 9.10, “Example of Resolving an Animal Location from Coordinates”

Note

If two or more sublocations contain the point, the one with the smallest area will be chosen.

- MovedOut event. Resolution will be the name of the Location associated with the Premises uri attribute of the MovedOut event.
- Provide the names of sublocations in a premises so that Premises editors can populate the pick list of Sighting location attribute editors

Figure 9.10. Example of Resolving an Animal Location from Coordinates



Chapter 10. Fair Editor

Overview

What is the Fair Editor?

The Fair Editor is a comprehensive open desktop solution for documenting, managing, reporting and exchanging information associated with livestock animal fairs and expositions. The Fair Editor is used for creating and editing documents that conform to the Fair information (data) model. Fair documents extend and compliment livestock animal history information held in Premises documents with fair exhibits, exhibitors, youth clubs, awards, and exhibitor parents information.

Project Goals

Livestock fairs and animal expositions are special kinds of livestock premises where the information associated with people, clubs, animals, exhibits, contest activities, and awards are recorded and managed. The goal of the Tracker Fair Editor is to create a comprehensive desktop based software application for Livestock fairs and expositions. An application that:

- imports external data from diverse sources
- adds interactive editors and views to record and visualize people and exhibit information
- acts as a data source to leverage powerful reporting tools for creating easy to understand reports

Extending the Tracker Premises livestock suite of tools to address animal livestock fairs and expositions is the primary goal of the Fair Editor project.

The Fair Editor provide models, editors, utilities, views and services for the creation and management of documents that:

- Offer bulk data input of exhibitor names, contacts, clubs and parents information
- Leverage full capabilities of the Tracker Premises Editor for managing animal life data and event histories for all animals that move in and out of the fair
- Model and visualize contact information for people associated with the fair, including exhibitors, supervisors, judges and parents of the exhibitors
- Model and visualize animal exhibits as the principle relationship between livestock animal life data, animal history, people contact information, contests and awards
- Record check-in of livestock animals from source livestock premises for early weigh in and at fair time
- Record weight and full livestock animal *life data*
- Offer comprehensive electronic interfacing of weight scales and RFID readers for efficient keyboard free inputing of data
- Dynamically calculate average daily gains of animal weights based on previous check-in weights
- Locate Animals that have moved to exhibits within sub-locations of a fair premises

- Resolve Animal locations that have moved to another premises after check-in or at the end of the fair.
- Assign animal exhibit positions based on visual sighting records or data inputs from industry standard GPS devices
- Act as a data source for a full featured Business Intelligence Reporting Tool BIRT for batch and ad-hoc report generation

Functional Overview

The Fair Editor is used to create documents that manage information for livestock fairs and expositions.

Note

The Fair Editor requires installation of the Premises Editor Feature.

If location visualization and location resolution services are desired, the Agriculture and Geography Features may be installed.

The Fair Editor extends the Tracker Premises Editor by integrating a new fair specific data model, views for managing people and exhibits, editors for creating documents, and utilities for importing people, youth club and exhibit information.

The Tracker Editor offers:

- Support for pre-fair preparation:
 - Utilities for importing tag numbers from external files
 - Utilities for importing people contact information from external files
- Support for pre-fair weigh-in:
 - Recording exhibitor information
 - Recording life data and the source premises of animals
 - Manual or electronic recording animal weights
 - Registering exhibitors for each animal
 - Printing registration summary reports for exhibitors
- Support for check in fair use-cases:
 - Simple tag scan to check-in animal
 - Manual or electronic recording animal fair registration weights
 - Automatic calculation of average daily weight gain based on pre-fair weigh-in
 - Assignment or reassignment exhibits to appropriate lots and/or classes
- Support for fair judging and awards
- Support for fair check out

Fair Documents

The Fair Editor enables users to create, transform and edit documents referencing animals, exhibits and people within a Premises.

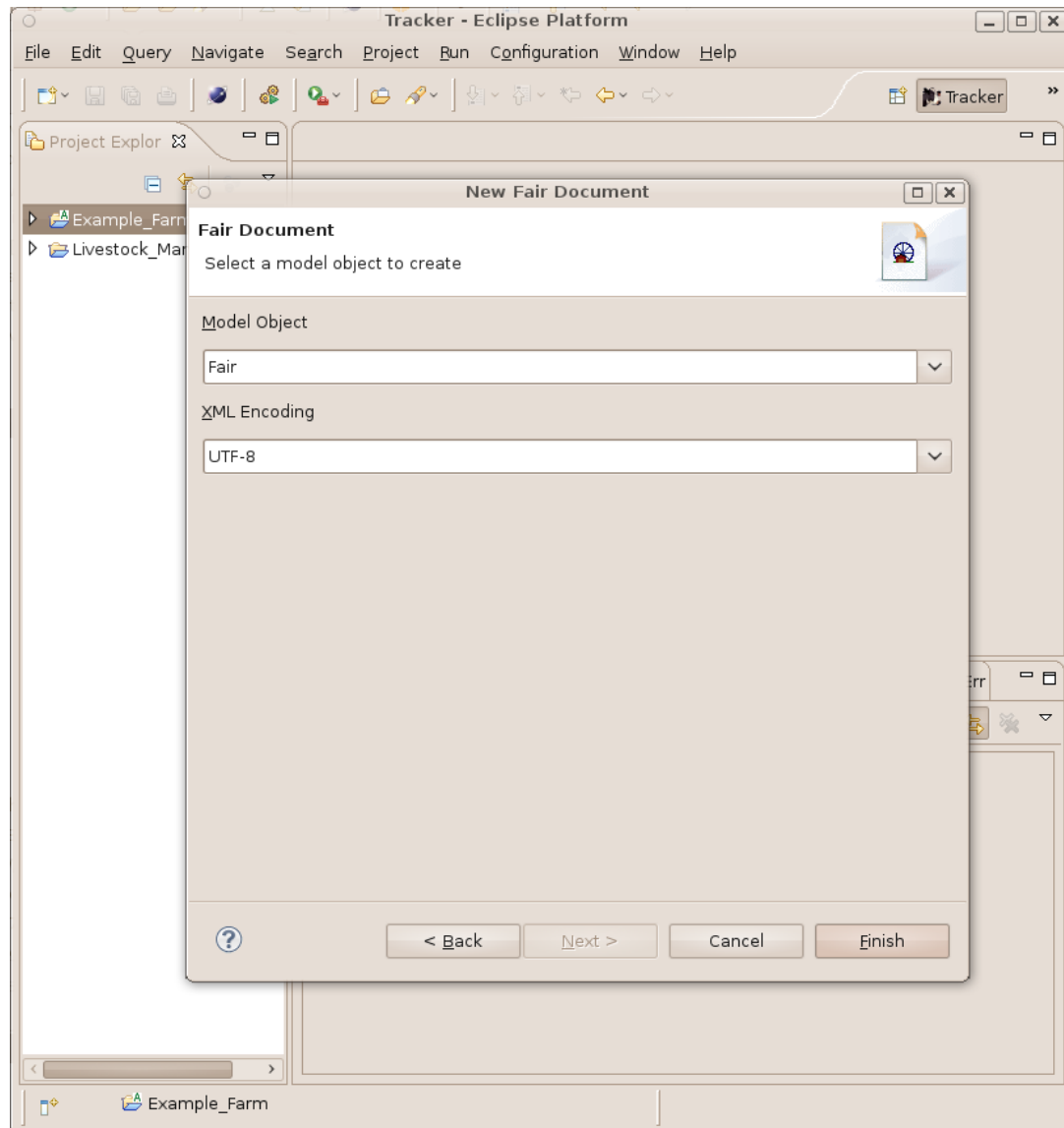
Fair document files utilize a `.fair` file extension to distinguish them from other types of files.

Editing Fair Documents

Creating a new Fair Document

To create a new Fair document. Go to the File menu and launch a *New Fair Document Wizard* by selecting the menu items: `File | New | Fair Document`.

Follow the instructions to choose a project, folder and a name for the document. Select `Fair` as the model object and press `Finish` to create the document.

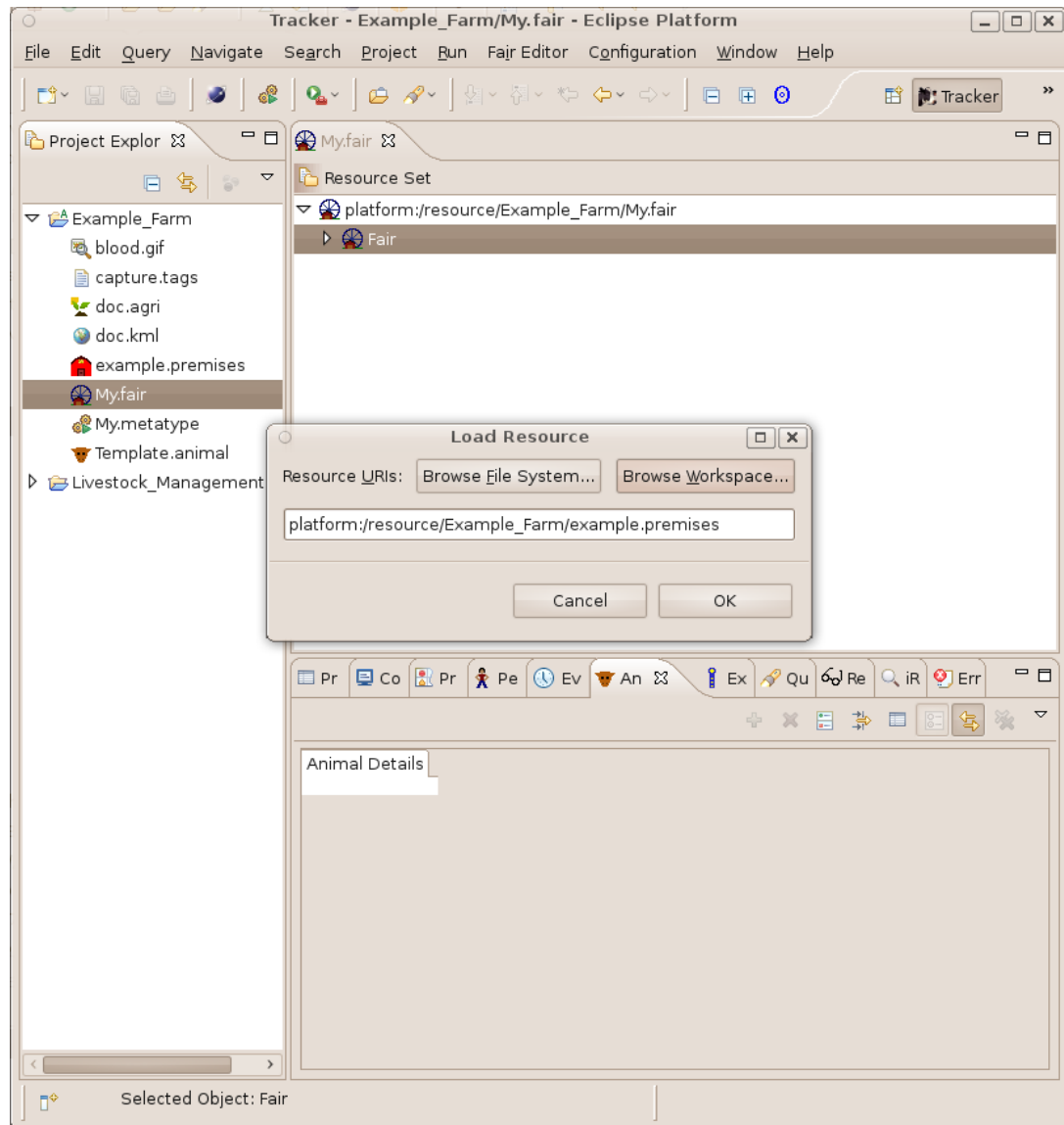
Figure 10.1. New Fair Document Wizard Model Object Prompt

Loading Additional Resources into the Document

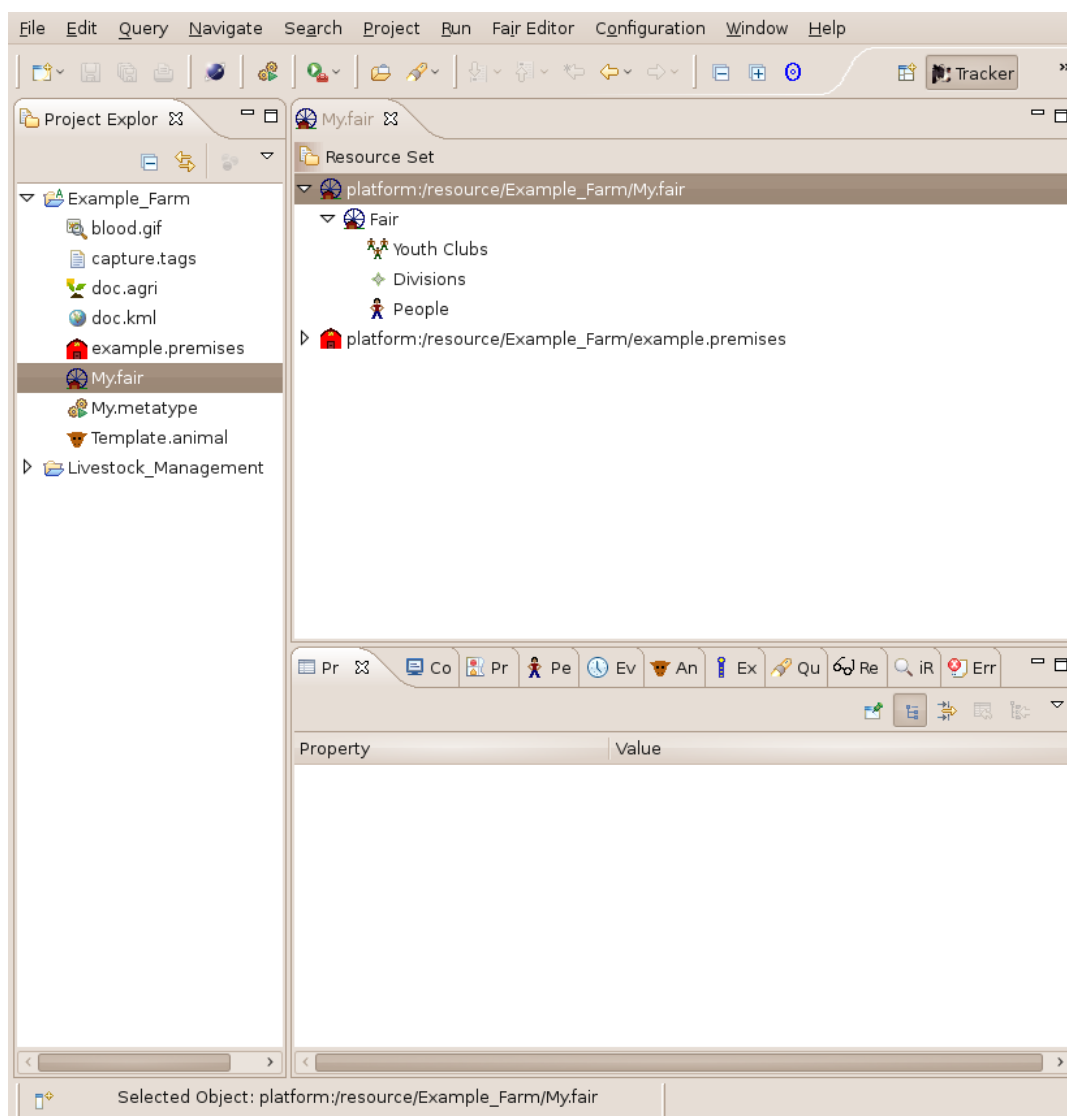
Note

Since the Fair document references the Premises element of a Tracker Premises document, a Premises document resources must already exist in the workspace. If it does not, create an empty Premises document first. *You can always add data to the referenced Premises document at the same time you edit the Fair document.*

To reference a Premises element from a Fair document, a Premises document must first be loaded into the Fair document as a resource. To load a Premises document as a resource place the mouse cursor on the root element in the Fair document, right click, and from the pull down menu choose Load Resource . . .

Figure 10.2. Load Resource Dialog

From the dialog choose the Premises document that you wish to load.

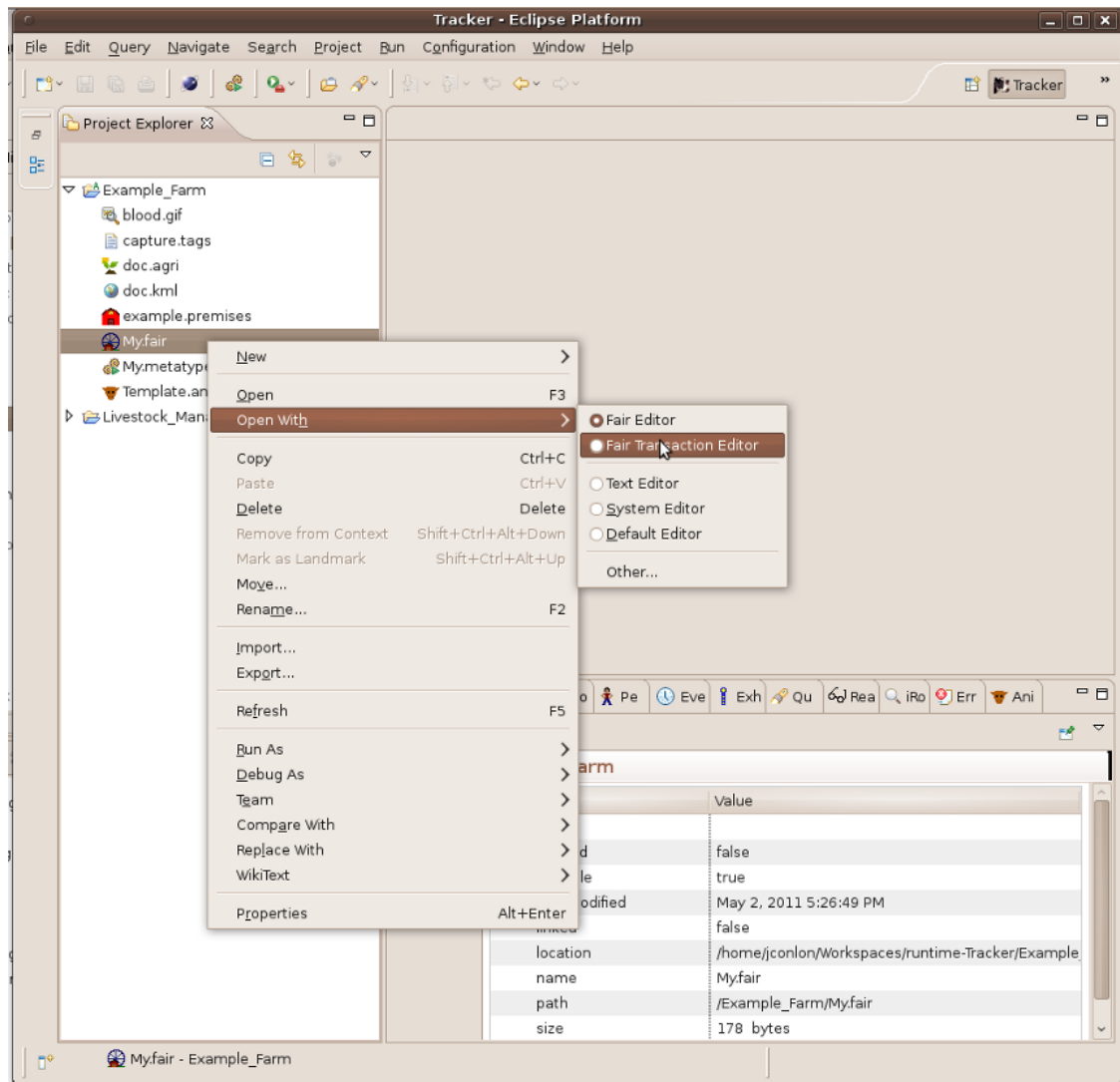
Figure 10.3. Fair Project with Loaded Resources

When Premises documents are loaded as a resource in a Fair document, the editor choice boxes associated with the Premises attribute of the Fair element will be shown as a choice for user selection.

Opening a Fair Document

For high throughput during fair weigh-ins and check-ins the Fair Transaction Editor can be used to edit fair documents. The Fair Transaction Editor supports the Tracker *Transaction* processing facility for electronic input of data originating from network connected sensors and RFID readers.

To open a Fair document with the Fair Transaction Editor, right click on the Fair document and select the pull down menu item *Open With*.

Figure 10.4. Opening a Fair Transaction Editor

Note

When not working with RFID readers or sensors, it is best to use the basic Fair Editor.

Fair Action Commands

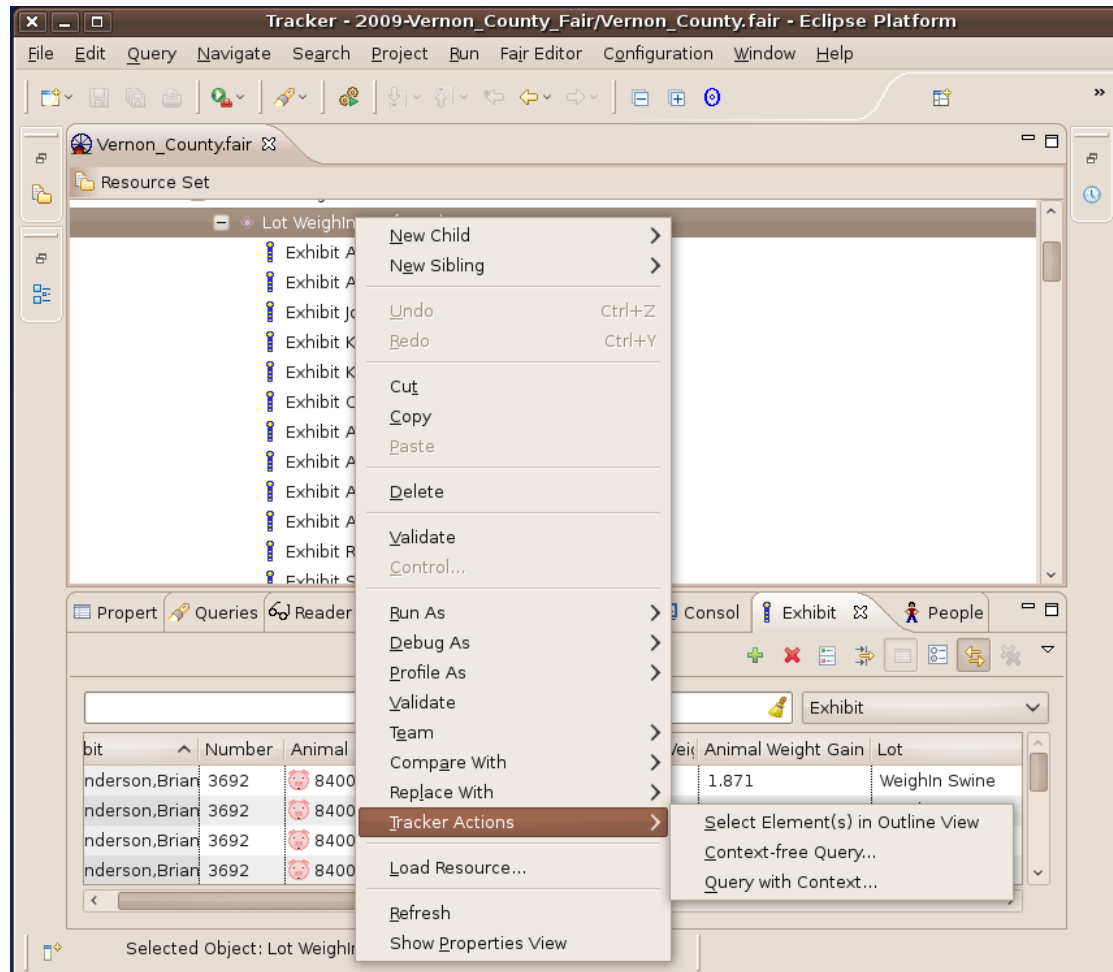
For key elements within the Fair document, and the associated Premises document the Fair Editor provides action commands that the user can invoke to effect changes on the content of the information held in these documents.

Fair Actions can be invoked on Fair document elements and for elements in the associated Premises document¹. These actions may be accessed as popup menus on the Fair and Premises document elements when viewing documents in the Fair Editor, document tree in the Outline View tree, the master table of

¹In addition to the actions that the Fair Editor adds to Premises elements the user will also see the actions that the Premises Editor and other provisioned features have added to the Premises elements.

the Exhibit View, the master table of the People View, the master table in the Animals View or the master table of the Events View.²

Figure 10.5. Invoking Element Actions from Fair Editor



² Right click on the element and the popup main menu will be shown. From there the sub menu 'Tracker Actions' will contain the appropriate action command.

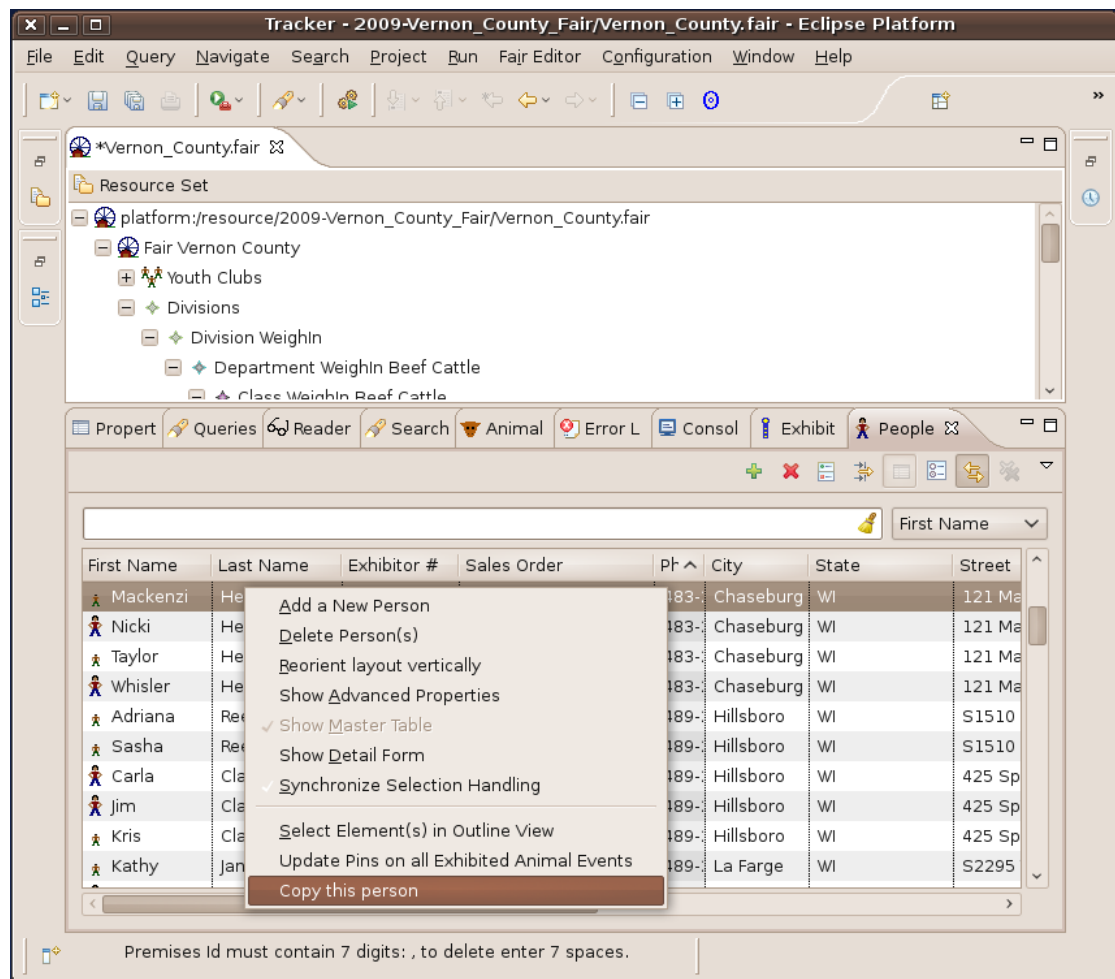
Figure 10.6. Invoking Element Actions from People View Master Table

Table 10.1. Action Commands on all Elements

| Element | Command Name | Fair Editor ^a | From People View | From Exhibits View | From Animals View | From Events View | Description | Comment |
|--------------|-----------------------------------|--------------------------|------------------|--------------------|-------------------|------------------|--------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| All elements | Select Element(s) in Outline View | Yes | Yes | Yes | Yes | Yes | Selects the same element or elements in the tree of the Outline View. | Helpful to sync from Views to the context of a tree view. |
| All elements | Context Free Query ^b | Yes | No | No | No | No | Shows a dialog for entry of a query to select sub (child) elements of the selection. | From a Fair or Premises element enter a query to find exhibits, people, animals or events. ^c |
| All elements | Query with Context ^d | Yes | No | No | No | No | Shows a dialog for entry of a query to select sub (child) elements of the selection. A combo box for selection of the query context. | Select the Fair or Premises and query to find exhibits, people, animals or events. |

^aThe Outline View like the Selection Page of the Premises Editor shows the document as a tree. All actions that can be accessed as popup menus in the Selection Page are also available from the Outline View.

^bOnly available when the OCL feature is provisioned.

^cSee the OCL Query documentation for more details.

^dOnly available when the OCL feature is provisioned.

Table 10.2. Action Commands on Person, Animals, and Lots

| Element | Command Name | Fair Editor ^a | From People View | From Exhibits View | From Animals View | From Events View | Description | Comment |
|-------------------------------------------------------|--------------------------------------------|--------------------------|------------------|--------------------|-------------------|------------------|------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|
| Person(s) | Update Pins on all Exhibited Animal Events | Yes | Yes | No | No | No | Persons pin value is set on all the moved events of his exhibited Animals | Helpful when the pin value is only known after the events take place. |
| Person(s) | Copy Person | Yes | Yes | No | No | No | Takes the information from one person element and duplicate it for a new person element. | Helpful when a family is registering to copy all addressing and parent information to a sibling. |
| Animal(s) with optional selections for Person and Lot | Register in Exhibit | Yes | No | No | Yes | Yes | Registers one or more animals in exhibits for a person. | Helpful during registration of exhibits. ^b |

^aThe Outline View like the Selection Page of the Premises Editor shows the document as a tree. All actions that can be accessed as popup menus in the Selection Page are also available from the Outline View.

^bOnce a person and their animals are added to the system, select the group of animals, the person, the lot to register them in, and invoke this command. It will create the exhibit. One or more animals are required for the selection. If the person or the lot is not included in the selection a wizard dialog will prompt the user for the missing information, either the person, the lot or both.

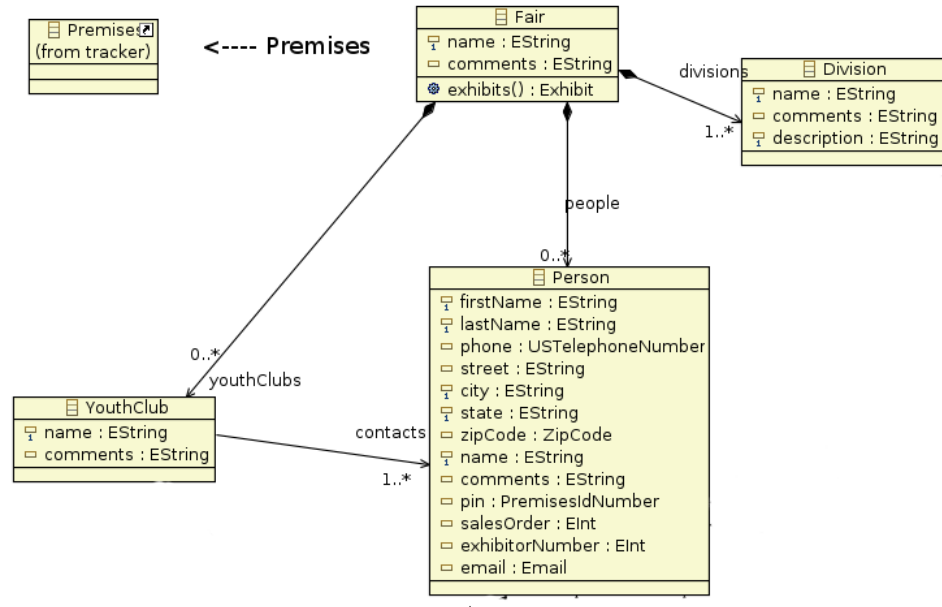
Fair Data Model

Because all user activity centers around the creation and manipulation of *Fair Documents*, understanding the structure (called the *data model*) of Fair documents is the easiest way to learn to use the information management facilities of the Fair Editor.

Fair Data Model Core Elements

All Fair documents conform to a very simple structure called the *Fair Data Model*. In the Fair Data Model, a single Fair element is given a name and a reference to a Tracker Premises file. Within this Fair element Youth Clubs, People, and Divisions of Exhibits are created by the operator.

Core elements of the Fair Data Model are shown in Figure 10.7, “Fair Core Data Model Elements”.

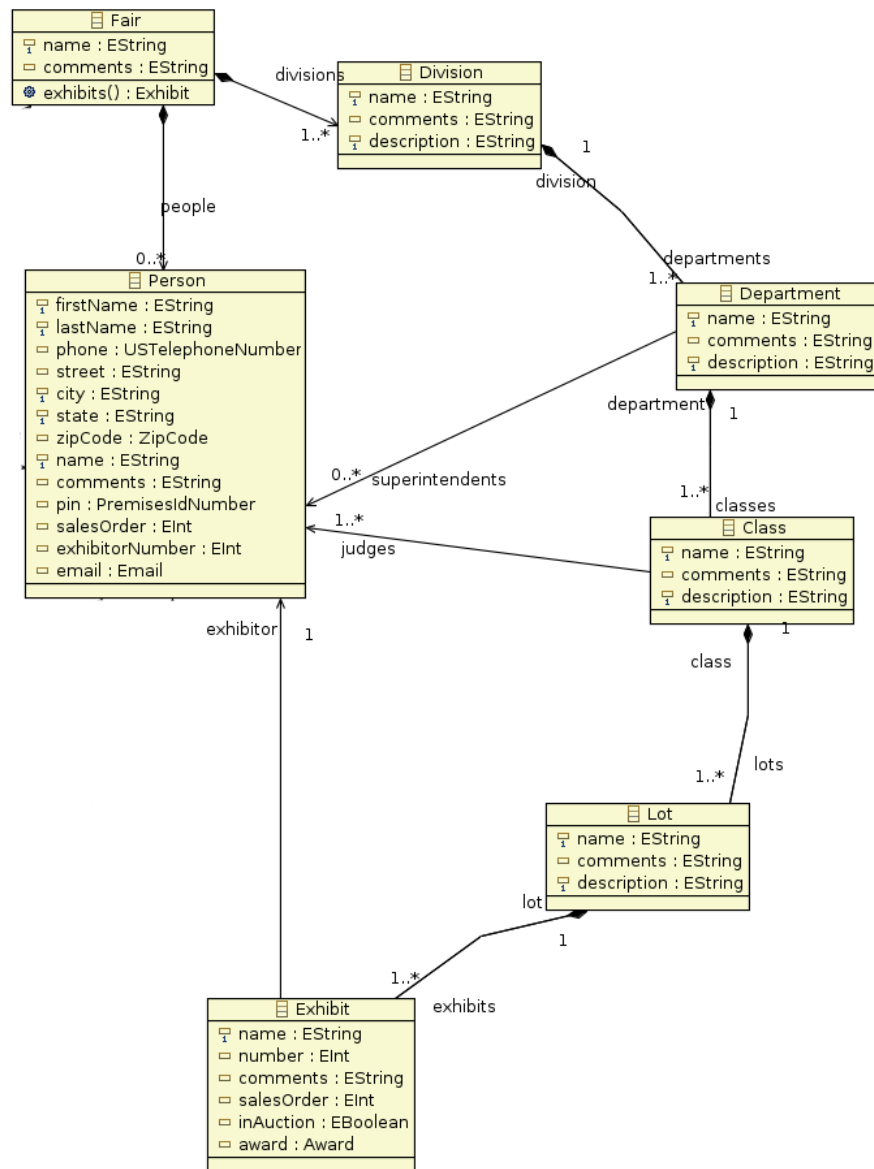
Figure 10.7. Fair Core Data Model Elements

- The core elements of the Core Model are depicted as rectangles in the figure.
- The notation within the elements are the attributes of the element.
- The notation between the elements are the relationships of the elements.

The following table offers a text explanation of the Element Relationships.

Table 10.3. Core Elements of the Premises Model

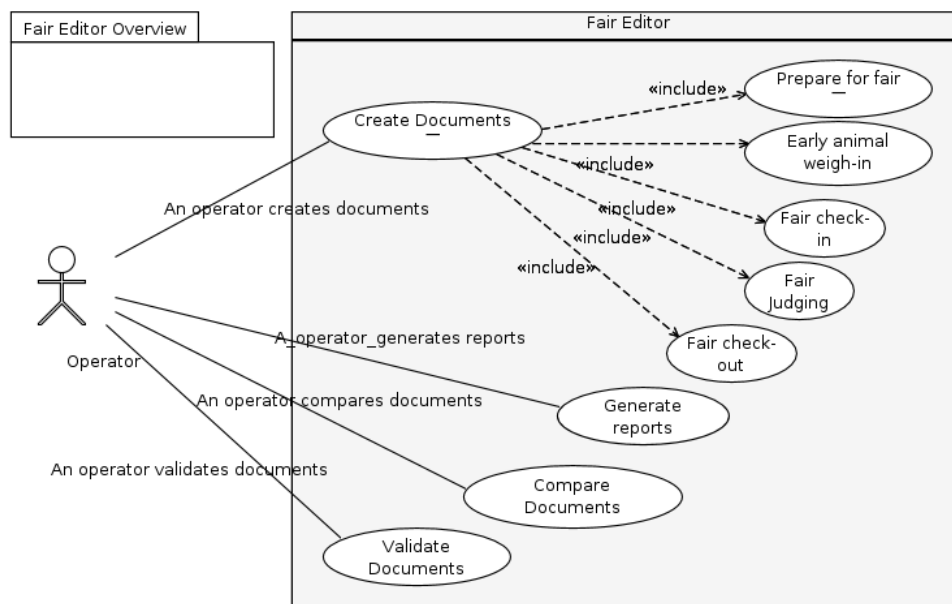
| Element Name | Relationship to other Elements |
|---------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fair | There is one Fair element. It acts as the first or root element of the document |
| name attribute | A Fair has a name that is assigned by the operator |
| premises attribute | A Fair has a premises attribute to reference the Premises document that will contain all the animal information |
| Person Elements | Fair has one or more person elements that are used to document people contact information. A fair document can specify people as exhibitors, parents, supervisors and judges. |
| Youth Club Elements | Exhibitors can be assigned to groups called Youth Clubs. |
| Exhibits | Exhibits are assigned within a hierarchy of Divisions, Departments, Classes and Lot containers. See the Figure 10.8, “Hierarchy of Exhibits” for the diagram showing how the hierarchy contains multiple child elements. |

Figure 10.8. Hierarchy of Exhibits

Fair Editor Use Cases

To best depict what the Fair Editor can do for Tracker users, the following figure presents an overview of the *Use-Cases* that can be performed.

Figure 10.9, “Fair Editor Overview” shows an overview of the usage requirements for the Fair Editor.

Figure 10.9. Fair Editor Overview

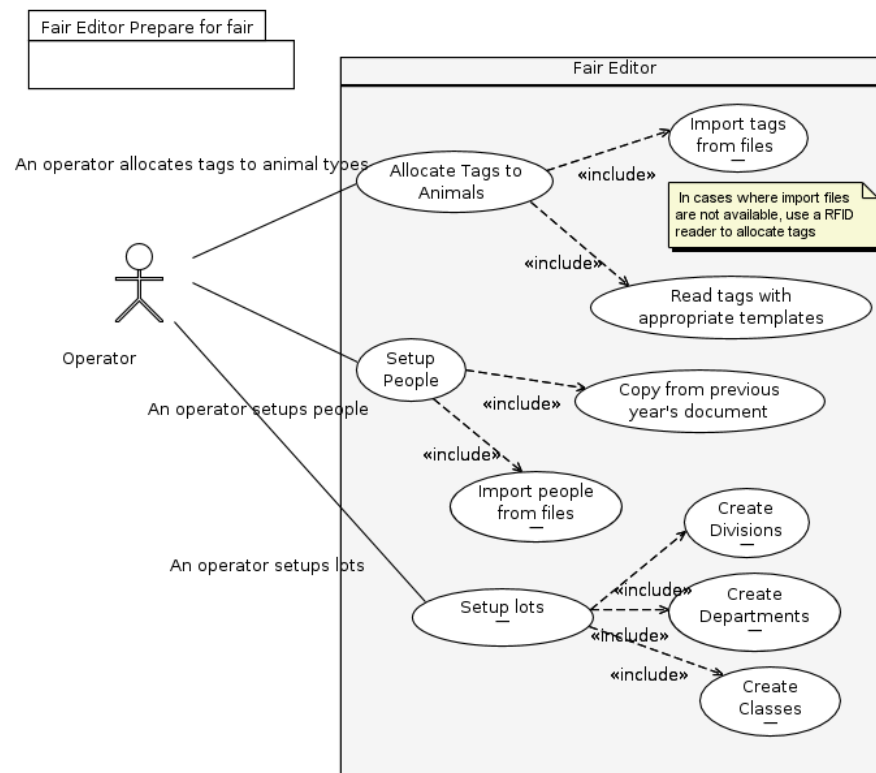
Note

Typically one Fair document and a referenced Premises document are the only two documents necessary for recording all uses-cases given for a fair. Youth club and person contacts can easily be copied and pasted from one year to the next.

Prefair Use Cases

Preparing for the fair is a simple matter of pre-allocating groups of tags to certain animal types, importing any people contact information that may already exist and setting up the types of contests that will make up the fair.

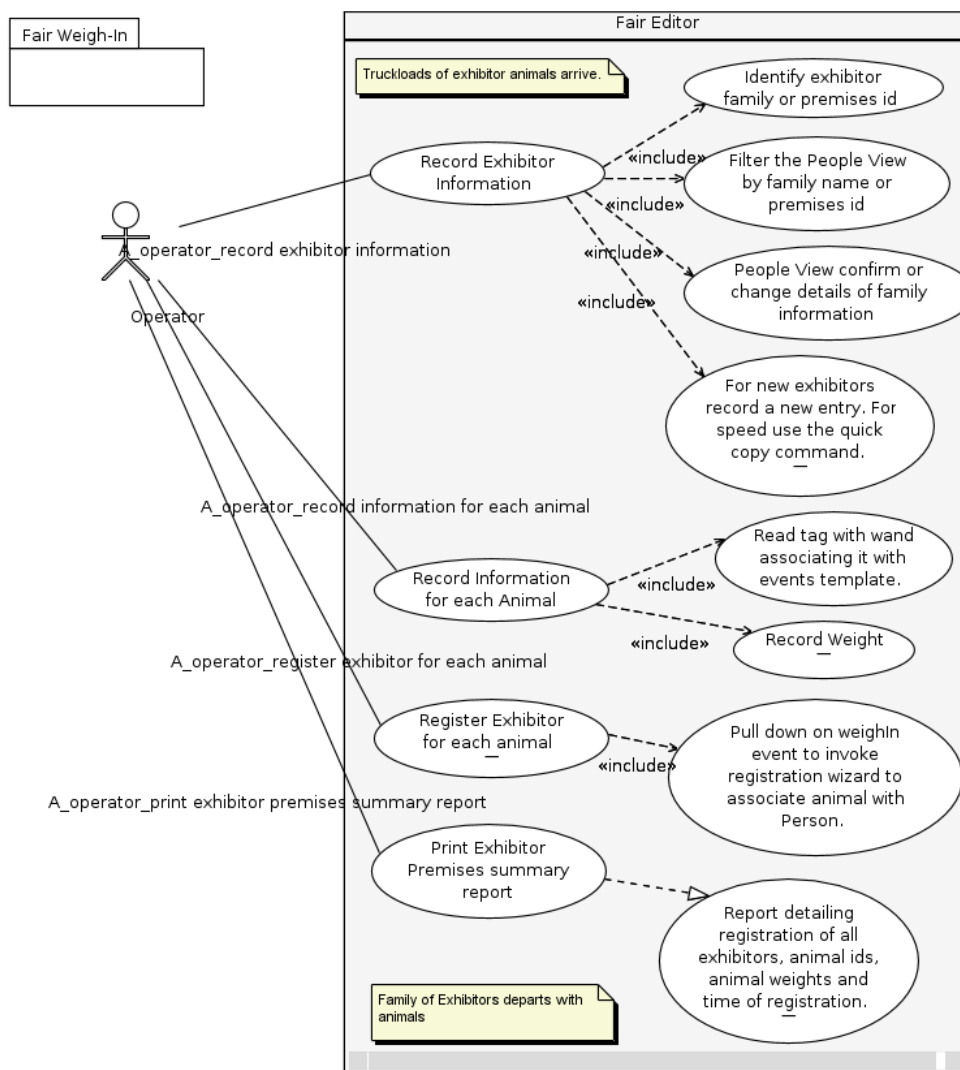
Figure 10.10, “Preparing for the fair” shows pre-fair usage requirements for the Fair Editor.

Figure 10.10. Preparing for the fair

WeighIn Use Cases

Many Fairs hold a early season weigh-in of animals in order to show an average daily weight gain.

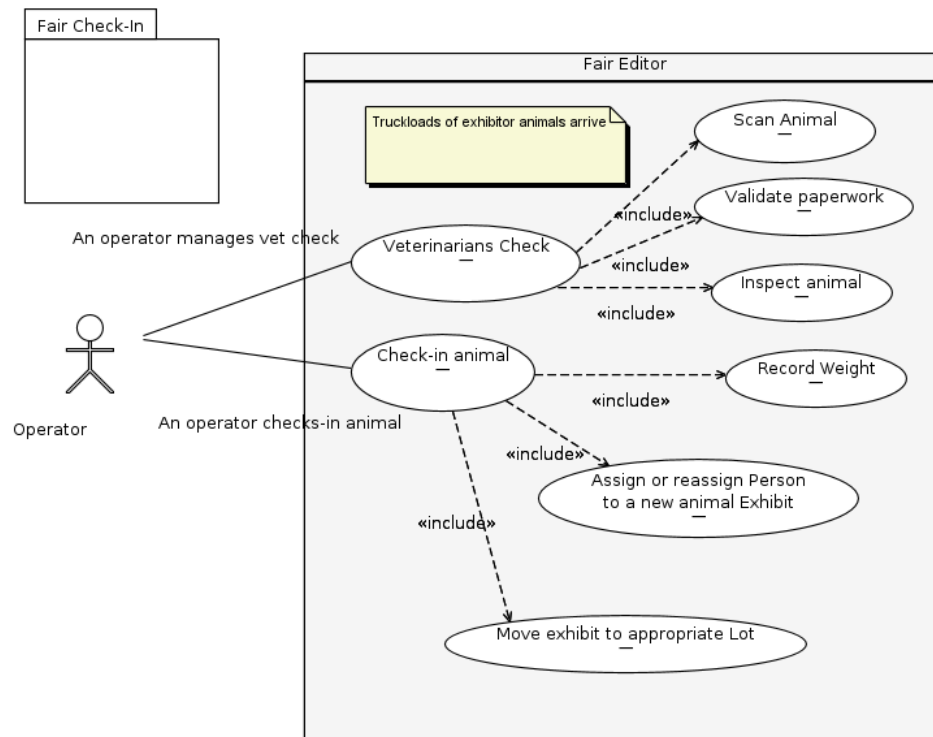
Figure 10.11, “Animal Weigh-In” shows early weigh-in usage requirements for the Fair Editor.

Figure 10.11. Animal Weigh-In

Check-In Use Cases

Checking in animals to the fair is often a two-fold process. First there is the veterinarian check to insure that the animal is healthy and has all required papers. Next the animal is checked in.

Figure 10.12, “Animal Check-In” shows early check-in usage requirements for the Fair Editor.

Figure 10.12. Animal Check-In

Chapter 11. Provisioned Features

Tracker Livestock Business Intelligence Toolkit Instances may be provisioned with one or more task specific tools or utilities to complement data model specific Editors.

The Tracker Business Intelligence Toolkit and Features consists of the following types of task specific utilities and tools:

- Data Importing Utilities;
- Data Exporting Utilities;
- Tools for Querying Operational Data;
- Utilities for interfacing with Tag Id Readers, including support for serial and Bluetooth connections;
- Utilities for interfacing with network attached sensors;
- Utilities for interfacing with industry standard Geography Information Systems GIS and geobrowsers;
- Eclipse Business Intelligence Reporting Tools BIRT.

Chapter 12. WLIC Export

Overview

What is WLIC Exporting?

The Wisconsin Livestock Identification Consortium WLIC is the State of Wisconsin's Premises and The WLIC is registering premises as the authorized agent of the Wisconsin Department of Agriculture, the Trade and Consumer Protection DATCP. WLIC is a multi-species effort led by Wisconsin's livestock and industry organizations. Contact WLIC at 1-888-808-1910.

The Wisconsin Livestock Premises Registration Act requires anyone who keeps, houses, or co-mingles livestock to register their premises.

Premises Document Export Functionality is provided by utilities that are called from the main Tracker Editor menu.

How to Export WLIC Information from Tracker Premises Documents?

Animal records submitted to the Wisconsin Livestock Identification Consortium WLIC utilize a specific file format.

The menu item *WLIC file...* will export the Tracker Premises Document in a WLIC file format.

Chapter 13. Fair XLS Import and Export

Fair XLS Import and Export Overview

Exporting Functionality

Warning

The functionality described in this chapter is provided on an interim basis only. It will be depreciated in the future when the Tracker reporting infrastructure is implemented.

Premises Document Export Functionality is provided by utilities that are called from the main Tracker Editor menu.

Exporting Information as Microsoft Excel documents

1. The menu item *Fair Registration XLS file...* will export the following Excel Spreadsheet information:

Table 13.1. Worksheets exported by the *Fair Registration XLS file...*

| WorkSheet Name | Properties exported | Comments |
|-------------------------------|----------------------------------------------------|------------------------------------------------------------------|
| Fair WeighIn and Registration | Parent, Club, Breed, WeighIn Date, Weight, Ear Tag | Combined listing focusing on the Registration and WeighIn Events |

2. The menu item *Fair Registration Raw Event XLS file...* will export the following Excel Spreadsheet information:

Table 13.2. Worksheets exported by *Fair Registration XLS file...*

| WorkSheet Name | Properties exported | Comments |
|--------------------------|----------------------------------------------------------|---------------------------------|
| Fair Registration Events | Ear Tag, Date, Participant, Address, Phone, Parent, Club | Fair registration event details |
| Weigh-In Events | Ear Tag, Date, Weight | WeighIn Event details |
| MovedIn Events | Ear Tag, Date, SourcePin | MovedIn Event details |
| MovedOut Events | Ear Tag, Date, DestinationPin | MovedOut Event details |

Chapter 14. OCL Query

Overview

Even though the structure of the Tracker Editor data models may be simple, some documents like Premises documents, can hold large amounts of animal and event history data. Obtaining information about or selecting specific animals or events in these large Premises documents require the use of advanced query and selection tools.

The *Object Constraint Language*, OCL is a simple but powerful declarative language used within Tracker's advanced query and selection tools to interpret Tracker Editor specific documents for information or to query documents in order to select specific elements for further manipulation.

Tip

When using the OCL Interpreter, Queries Viewer, or Query tools to interrogate Premises documents refer to the Editor specific data model when composing OCL query statements.

OCL Interpreter

The OCL Interpreter is a parser/interpreter¹ for the Object Constraint Language (OCL) version 2.0 for EMF. Using this parser, you can evaluate OCL expressions on elements in any Tracker document.

To bring up the OCL Interpreter, from the Tracker Editor menu in the Workbench:

- a. From the main menu select the *Query* menu. See Figure 14.1, “Invoking OCL Console”.
- b. Press the *Show Console* menu item.

The OCL Interpreter console will show below the Tracker Editor. See Figure 14.2, “OCL Query Console”.

¹Tracker's OCL interpreter is based on the work done in the *EMF OCL Tutorial* [<http://help.eclipse.org/help32/index.jsp?topic=/org.eclipse.emf.ocf.doc/tutorials/oclInterpreterTutorial.html>].

Figure 14.1. Invoking OCL Console

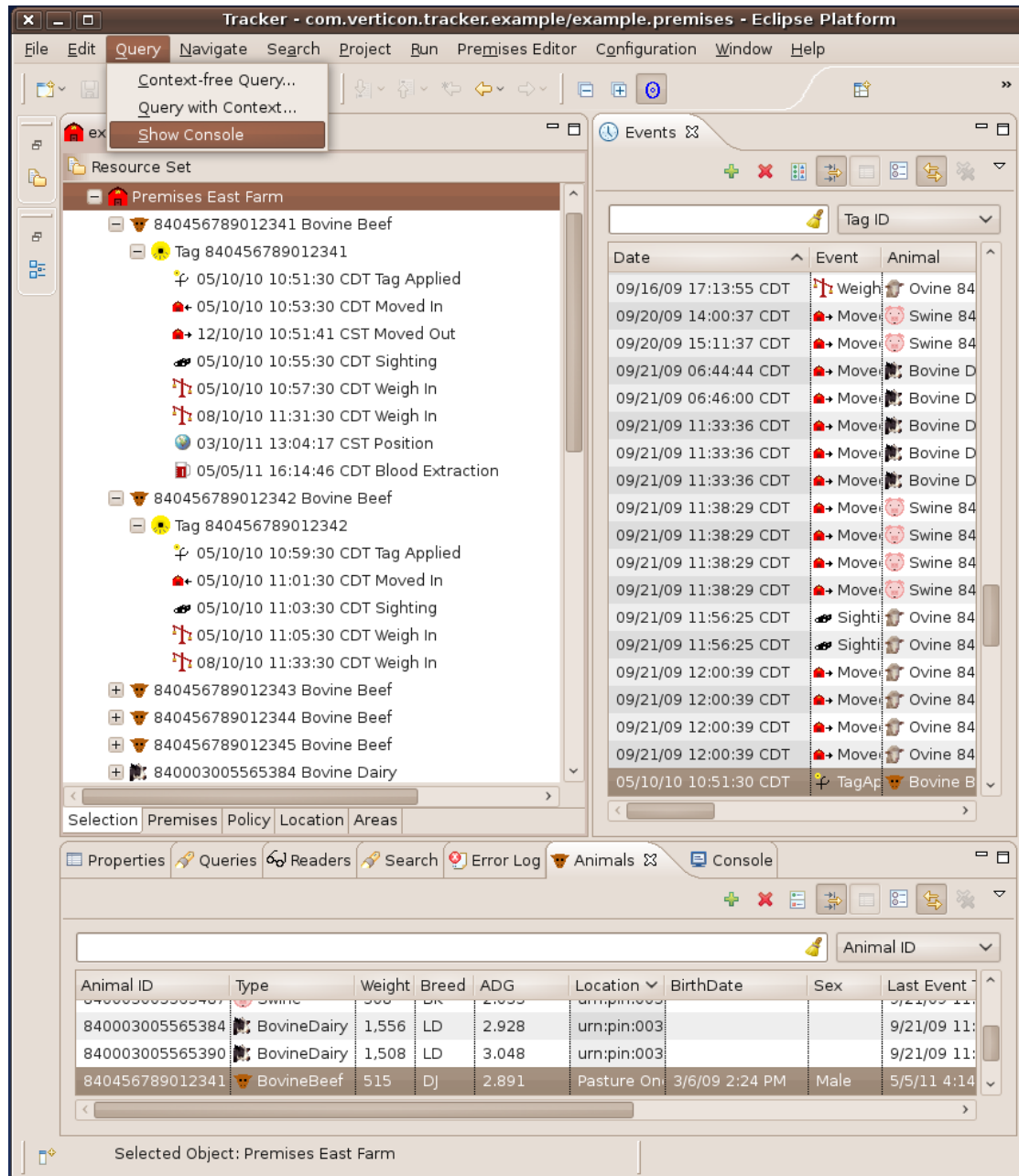
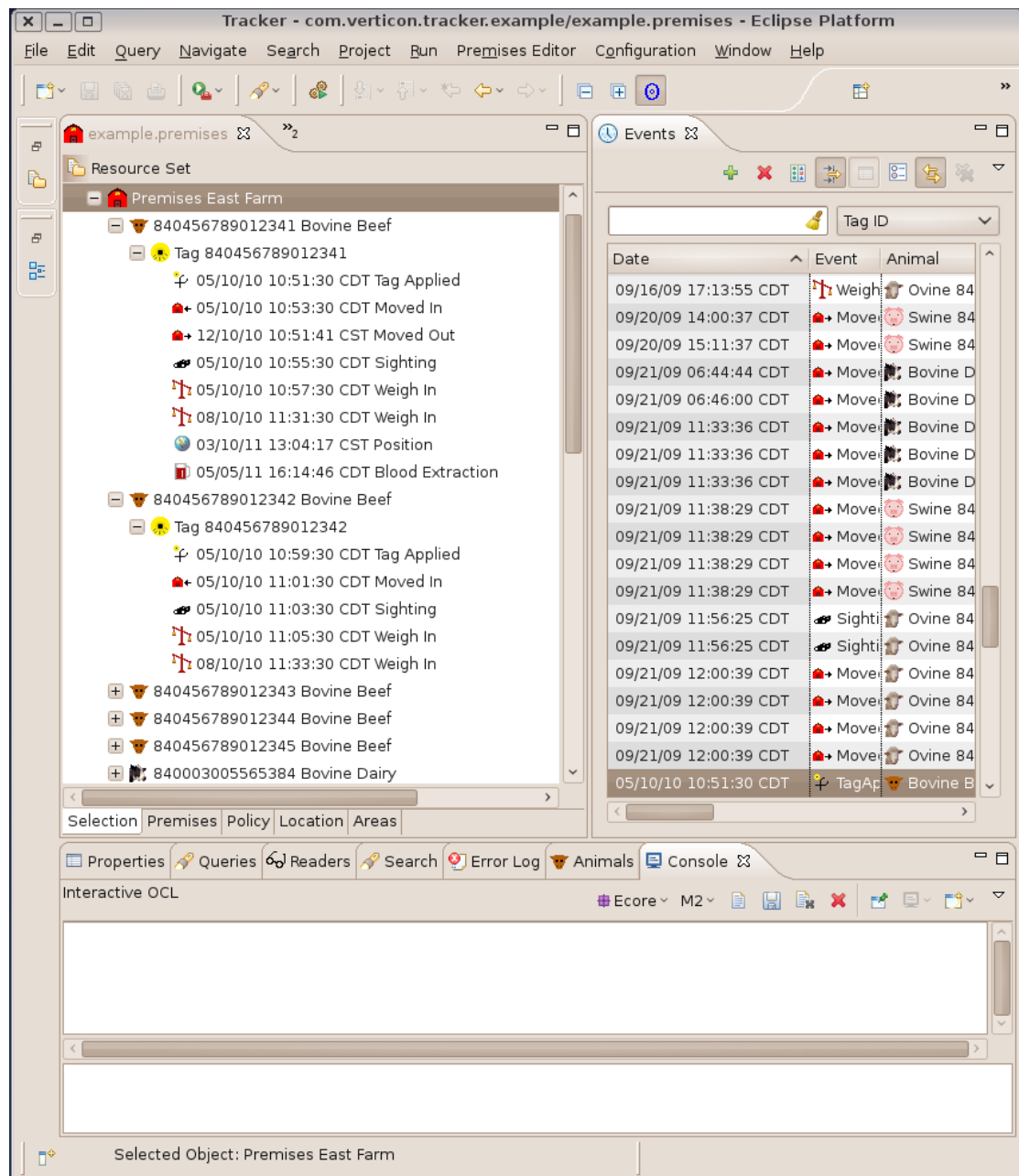


Figure 14.2. OCL Query Console



The following are example OCL Interpreter rules that can be used for the querying Premises documents. All examples use the `Premises` Element as the *Context* for Interpreter. Select the Premises Element in the Premises Document and cut and past the following examples into the lower pane of the OCL Interpreter Console Window.

Example 14.1. Listing of animals and the number of events associated with each.

```
animals -> collect(  
  a: Animal | Tuple {  
    animal: Animal = a,  
    numEvents: Integer = a->size()  
  }  
)
```

Example 14.2. Listing of animals and the set of events associated with each.

```
animals -> collect(  
  a: Animal | Tuple {  
    ain: String = a.id,  
    events: Set(Event) = a.allEvents()  
  }  
)
```

Example 14.3. Listing of animalIds and the WeighIn events associated with each.

```
animals -> collect(  
  a: Animal | Tuple {  
    ain: String = a.id,  
    weighIns: Set(WeighIn) = WeighIn.allInstances()->select(id=a.id)  
  }  
)
```

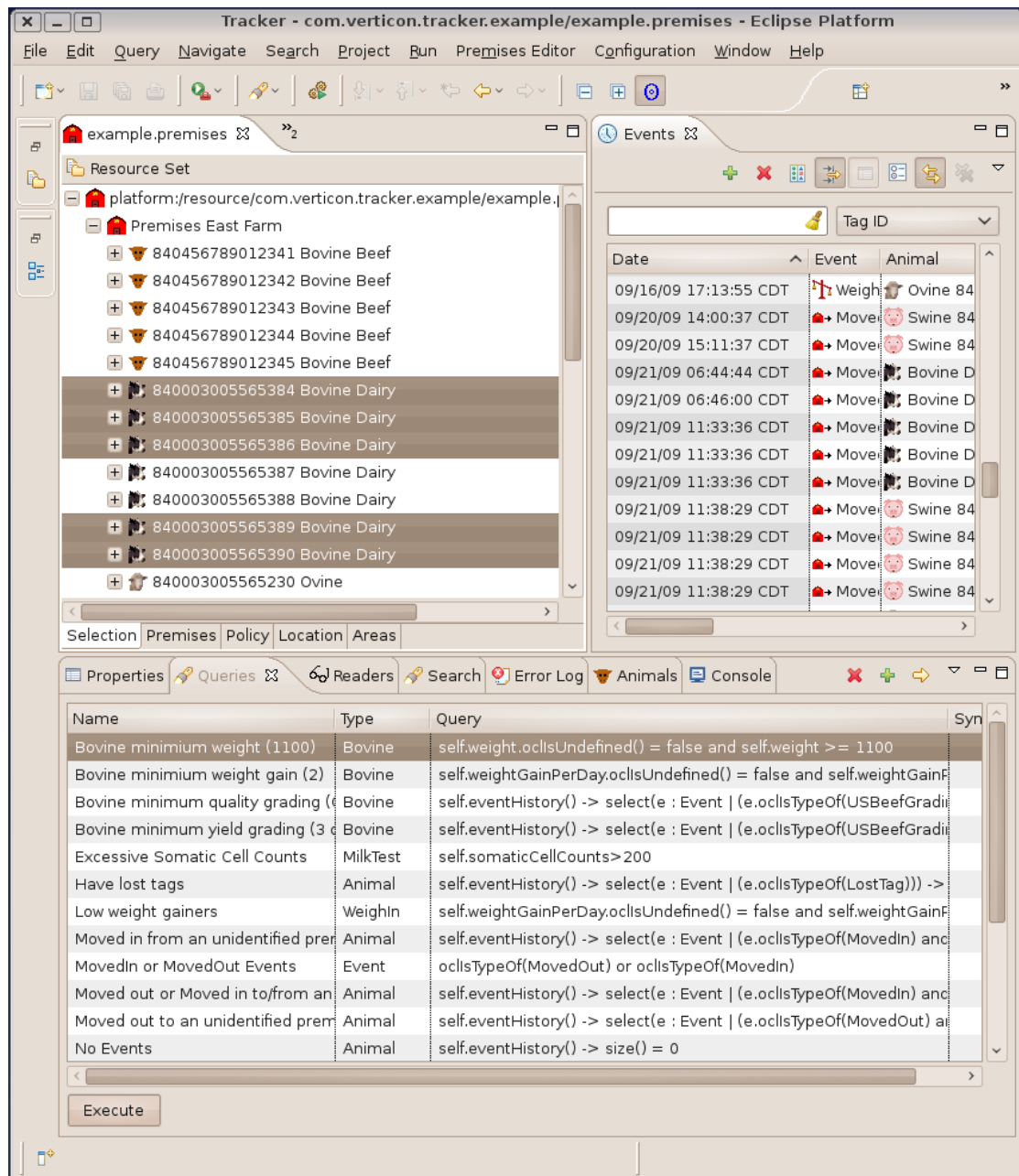
Example 14.4. Listing of animalIds and the Weights associated with each.

```
animals -> collect(  
  a: Animal | Tuple {  
    ain: String = a.id,  
    weights: Set(Integer) = WeighIn.allInstances()  
      ->select(id=a.id)->collect(wi:WeighIn | wi.weight)  
      ->asSet()  
  }  
)
```

Queries View

The Queries View is a convenient way to invoke frequently used OCL queries to select specific elements within Premises documents.

Figure 14.3. Queries View



In order to use the Queries View you will have to configure the viewer by adding individual queries. The view will persist all queries between boots of the Tracker application, so you can over time create a suite of personalized custom queries associated with how you wish to select elements. Once the queries are added to the view using them is as simple as selecting them and pressing the *Execute* button.

To add a custom query to the Queries Viewer, select the Queries Viewer tab and press the Add button. A new row representing a OCL query will be added to the Queries Viewer Table. Each OCL Query row contains four columns.

- Name column takes an arbitrary string name used to identify your query. You will want to use a descriptive name here, so you can find it latter in order to invoke it quickly.

- b. Type column is the element you wish to use as the target for your query.
- c. Query column is the OCL query string itself. See following section for examples.
- d. Syntax Errors column is where erroneous Query strings will output error messages. Once you enter a correct query string in the Query column you will not see any error messages in this column.

The following are a few example Query strings that can be used for adding custom OCL Queries to the Queries Viewer.

Example 14.5. Find the animal with active tag 985152002153189.

Target Animal

Query -

```
self.id='985152002153189'
```

Example 14.6. Find all MilkTest Events with Somatic Cell Counts > 200

Target MilkTest

Query -

```
self.somaticCellCounts>200
```

Example 14.7. Find all Animals with undefined MovedIn Source Pin attributes

Target Animal

Query -

```
Event.allInstances()->
  select(e : Event | e.id=self.id and e.ocIsTypeOf(MovedIn) and
    e.ocAsType(MovedIn).sourcePin.ocIsUndefined() )->size()>0
```

Chapter 15. Tag ID Reader

Tag ID Reader Overview

Adding Event Transactions to Documents

Tracker provides two types of editors for working with Premises documents. For electronic input of data via connections to Tag Readers a Transaction editor is used. The Premises Transaction editor can connect to one or more RFID Tag Readers. As animal tags are read by connected Tag readers the tag ids will be added by Transaction editors to the document as animal history events for the animal identified by the tag ID.

Note

Tracker does not require the use of RFID Tag Readers or special tag ID types to create and or manage animal history events. History events can easily be added manually or imported from other files. If no electronic reading of tags is required just use the non-transaction based Premises Editor.

Similar to Tracker Bulk Event Processing, Tracker Readers use Animal Document Templates to *associate* event histories with specific RFID reader tag IDs. As RFID reader tag IDs are received they are published as one or more history events of the Animal corresponding to the Tag/@idNumber.

Currently there are four types of connections to Readers supported by the Tracker.

- a. Bluetooth Reader - connects to Bluetooth RFID tag readers;
- b. Serial Port Reader - connects to Serial Port RFID tag readers;
- c. Generating Reader - generates random test tags from a file. (For testing only);
- d. File Reader - reads Tag/@idNumbers as they are appended to a capture file.

File Reader

In most cases Readers will use Bluetooth or Serial Port connections. But in cases where non-standard reader connections or audit records are required a File Reader can be used.

Tag readers transmit the numbers representing the animal id as an ASCII text message. A generic way to handle connections to RFID devices is to use a terminal emulator program to connect the Tag reader and capture output from the connection to a file. As tags are read by the reader the tag IDs will be appended to the end of the file.

Tracker's File Reader can be configured to periodically read tag IDs added to the tail end of a specified text file.

Usage and operations

Tracker Reader View

To show the Readers View. From the Workbench Window menu

- a. Select Show View
- b. Select Other
- c. Select Tracker
- d. Select Readers

Typical premises projects contain a premises document and one or more animal template documents. To begin, make sure your project contains one or more animal template documents. Next open the premises document with the Premises Transaction editor. This document and Transaction Editor are now set as the active editor and active document. In the Readers view invoke the *Add Readers Wizard* by selecting the plus symbol in the view toolbar. A list of all the animal templates in your active project will be shown. Pick one and continue to follow the prompts in the wizard to add the kind of reader you require.

Global Configuration of File Reader Timers

To change the timing of parameters of all the configured File Readers. From the Workbench window menu:

- a. Select Preferences...
- b. Select Tracker
- c. Select FileReader

Configuring requires setting the preferences for:

The `Capture File Read Interval` is the time interval to wait before rereading the file for new messages.

Chapter 16. iRouter Premises Gateway

Many animal premises managers require the integration of sensors, actuators, weigh scales, and RFID readers. Traditional solutions are often custom software and network integration solutions that require business analysts, programmers and installers to undertake a costly and time consuming development, integration and deployment project. Verticon's iRouter is a game changing alternative to high cost project integration efforts. The iRouter is a Just-In-Time integration solution for multiple input devices, sensors and actuators.

The Premises Gateway feature is an iRouter feature that facilitates the bidirectional movement of information between *iRouter* network attached sensors, actuators, business logic services and *Tracker Premises Documents*.

Note

See the iRouter User Guide [<http://tracker.verticon.com/Tracker/documentation/iRouter.pdf>] for details about the iRouter.

The Premises Gateway feature contains three iRouter Component Factories and an Gateway EventHandler.

Table 16.1. Premises Gateway Component Factories

| Component Factory Name | Description | Consumes | Interface input | Produces | Interface output | |
|------------------------------------|-----------------------------------------------------------------------------------------------------------|------------------------------------------------------------|-----------------------------------------------|-----------------------------------------------------------------------------------------|------------------------------------------------------|--|
| Premises EID Reader Gateway | Gateway from EID Tag Readers to the iRouter network. | | Subscribes to EventAdmin RFID Tag Read Events | EID Number Products | | |
| Premises Measurement Event Gateway | Gateway from the iRouter that creates Measurement and Position Events in Premises documents. ^a | Consumes Measurement, Position and optional State Products | | | Publishes EventAdmin Measurement and Position Events | |
| Animal Life Data Gateway | Gateway from Premises document Animal elements to the iRouter network. | | The Animal elements in a Premises document | List of Strings as a Product, where each string represents the life data for an animal. | | |

^aThis function is performed by the Gateway EventHandler.

Premises EID Reader Gateway

com.verticon.tracker.irouter.eid.event

Many livestock information management solutions require the integration of one or more RFID Tag Readers with network attached sensors and actuators. The EID Reader Gateway Factory is a component factory for creating services that interface with RFID Tag Readers integrating their Tag reads with iRouter network services.

The EID Reader Gateway Factory services transform *EID* numbers read from *RFID* Tag Readers and produce iRouter *EID number* products for consumption by iRouter Consumer services..

RFID/EID Reader Background

The Tracker Reader Feature and the Tracker Premises transaction editor provide the functionality for configuration, management, dynamic editing and interfacing of *RFID* tag readers and Premises documents.

Opening a Premises document with the Premises Transaction Editor will enable input connectivity with all the connected Tag Readers in the Readers view. During this transaction editing session when tags are read by a Tag reader, the EID numbers transmitted by the Tag Reader will be converted to one or more Animal history events, and added to the Premises document. The operator will see this as a dynamic change to the document.

See the `com.verticon.tracker.reader` feature [http://tracker-project.verticon.com/verticon_Tracker/admin/ticket/components/com.verticon.tracker.reader] and the Tracker Business Intelligence Toolkit documentation for more details.

EID Number Events to iRouter EID Products

Readers view connected Tag Readers publish EID numbers on the EventAdmin Service the internal inter-process communication facility.

The EID Reader Gateway Components are gateways that subscribe to EID numbers published on the EventAdmin Service and re-produce them as *EID number* products on the iRouter WireAdmin service.

To provide the gateway functionality, an EID Component implements `EventAdmin EventHandler` and `WireAdmin Producer` services that:

- Listen for Tag reader EventAdmin EID events that are generated by one or more Tag Readers
- Transform the EID events coming from Tag Readers into iRouter *EID number* products for processing by iRouter consumer services.

Premises Measurement Event Gateway

com.verticon.tracker.irouter.measurement.event

The Measurement Event Gateway Factory is a factory for creating gateway services that consume iRouter *Measurement* and *Position* products and transform them into EventAdmin Measurement and Position Events.

iRouter Measurement and Position Products to EventAdmin Events

To provide the gateway functionality, a Measurement Event component implements the `WireAdminConsumer` service to consume:

- an iRouter *Measurement* or a *Position* product
- an *EID number* product, which can come from an EID Reader Gateway component (mentioned above) or from any iRouter producer of *EID number* products.
- an optional *State* product

The gateway caches the last received *Measurement* and *Position* object until a user configured triggering condition occurs, then it sends out the cached product with an ID value as event to the EventAdmin Service. The ID value is determined from:

- ID implicit to the cached product
- *EID number* product explicitly received and cached by the gateway

Two triggering conditions can be configured by the user

1. *EID number* product
2. *State* product

Gateway EventHandler

The Gateway EventHandler is a gateway service that transforms Measurement and Position events coming from the iRouter into Premises animal history events.

Animal History Events

Tracker Editor Feature provides a Tracker Transaction Editor for creating animal event history events from *EID* tag readers.

See the `com.verticon.tracker.editor` feature [http://tracker-project.verticon.com/verticon_Tracker/admin/ticket/components/com.verticon.tracker.editor] and the Tracker Business Intelligence Toolkit documentation for more details.

EventAdmin Events to Animal History Events

The Premises Gateway implements `EventAdmin EventHandler` to listen for EventAdmin Events coming from the *Measurement Event Gateway* (mentioned above). The Premises Gateway transforms these events to Premises animal history events.

- iRouter *Measurement* products are transformed into animal history *weighIn* events.
- iRouter *Position* products are transformed into animal history *position* events.
- All other SI [<http://en.wikipedia.org/wiki/SI>] measurements originating from iRouter network attached physical sensors or internal iRouter business logic services can be saved as a *Generic Animal History Event* in a Tracker Premises documents. For these measurement events a `Metatype` `OCD` will have to be configured and its ID specified in the configuration of the Measurement Event Gateway service.

Animal Life Data Gateway

com.verticon.tracker.irouter.animal.data

Premises documents contain both animal history and animal life data. For external devices that require synchronization of animal life data, the Animal Life Data Gateway Factory creates services that produce a list of animal life data records extracted from a specified Premises document. These animal life data record products can be consumed by iRouter services to export animal life data to external devices.

- Extraction process is triggered when a consumer connects to this service
- Data sent as a list of records containing specified animal attribute values
- Reference premises document with URI
- User configures:
 - URI of premises document
 - Which animal attributes to use as fields
 - Which order to place the fields
 - Optional transformation of attribute values to field values
 - Optional static values inserted as fields
- Refresh time before a document is reloaded and new records are created

Premises Gateway Examples

The iRouter and Tracker technology configurations in this section are simple solution examples for solving common livestock premises integration problems.

Recording Animal Location Example

An example of a simple Premises Gateway integration solution is the Figure 16.1, “GPS Gateway Example”. This is an iRouter Monitor view image snapshot of a GPS unit integrated with RFID readers. The purpose is to record the locations of scanned animals as Premises document Position events.

This iRouter configuration example consists of:

- A Bluetooth gps receiver
- An iRouter GPS Proxy component
- Three RFID Tag Readers configured in the Readers view
- iRouter EID Gateway component configured to listen to EID tag reads from all of the Tag Readers
- iRouter Measurement Event Gateway that creates position events when a tag is read
- iRouter Gateway EventHandler that adds position events to the active premises document

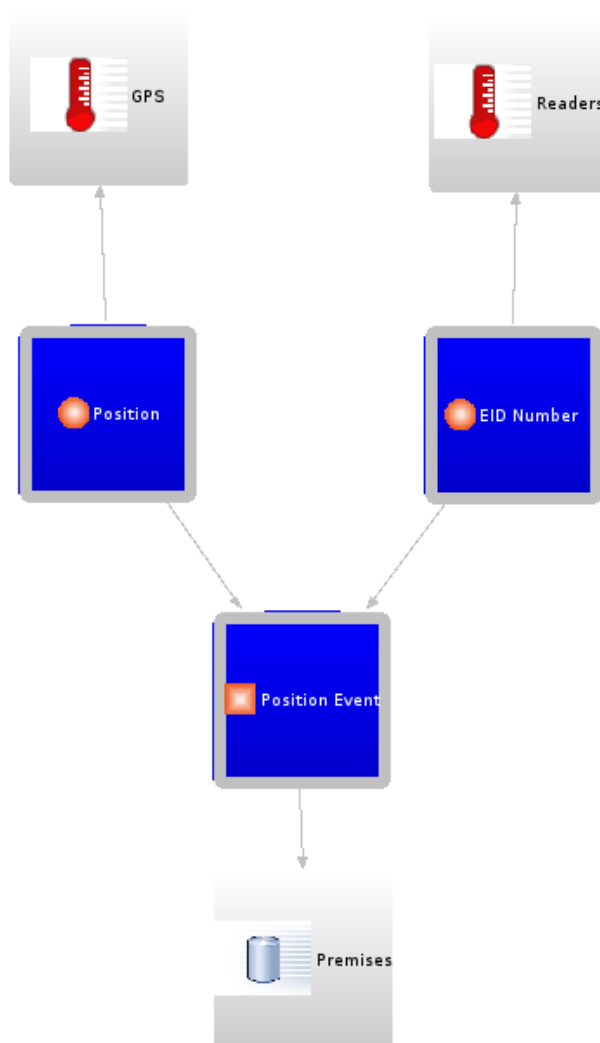
When this configuration is loaded into a Tracker Desktop and a premises transaction editing session is active; *any operator can record animal Position Events by scanning animals with any one of the three configured RFID readers.*

Note

Multiple Readers and multiple GPS sensors can be configured to create a comprehensive geo-location aware premises for operator animal location resolution. For more information about configuring iRouter sensors and actuators and displaying active configurations with the iRouter configuration view see the iRouter User Guide [<http://tracker.verticon.com/Tracker/documentation/iRouter.pdf>].

See the Agriculture Feature for details of Tracker's geo-location resolution solution.

Figure 16.1. GPS Gateway Example



Recording Animal Weight Example

An example of a more complicated Premises Gateway integration solution is the Figure 16.2, "Animal Weight Gateway Example". This is an iRouter Monitor view image snapshot of a TruTest™ scalehead indicator integrated with RFID readers. The purpose is to

- synchronize the TruTest indicator with the animal life data from the active Premises document,
- initialize animal weighing session on the indicator by scanning an animal tag,
- record the weights of animals as Premises document WeighIn events when an operator presses the indicators *ENTER* key.

The configuration consists of:

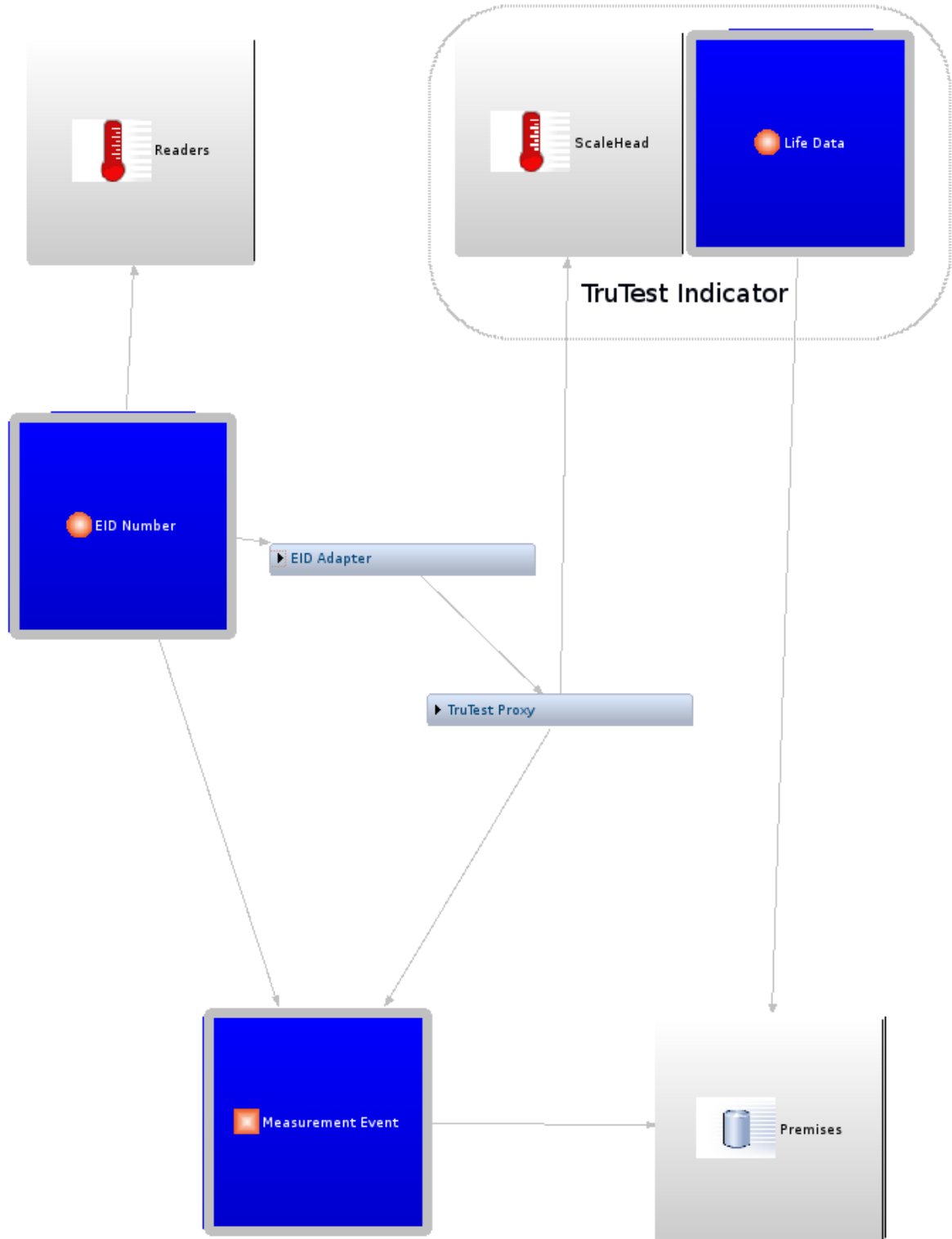
- ATruTest™ scalehead indicator and weigh bars
- TruTest™ proxy component
- TruTest™ business logic adapter components
- Animal Life Data synchronization components
- Three RFID Tag Readers configured in the Readers view
- iRouter EID Gateway component configured to listen to EID tag reads from all of the Tag Readers
- iRouter Measurement Event Gateway that sends weight and EID when a operator presses the enter key on the scalehead
- iRouter Gateway EventHandler that adds weighIn events to the active premises document

When this configuration is loaded into a Tracker Desktop and a premises transaction editing session is active; *any operator can scan animals with any one of the three configured RFID readers* to identify the animal to the scalehead indicator. When an operator presses the *ENTER* key on the indicator, a weighIn event, will be added to the scanned animal's event history on the Premises document.

Note

Multiple Readers and multiple scaleheads can be configured to create a comprehensive multi station weighIn facility for animals. For more information about configuring iRouter sensors and actuators and displaying active configurations with the iRouter configuration view see the iRouter User Guide [<http://tracker.verticon.com/Tracker/documentation/iRouter.pdf>].

See the TruTest Feature for details of iRouter TruTest™ integration solution.

Figure 16.2. Animal Weight Gateway Example

Chapter 17. iRouter Premises Common Producers

Tracker Desktop can be used by small to medium Livestock producers in an on-line production capacity, but one or more desktop (or laptop) may not offer a complete solution for larger enterprise operations.

Enterprise livestock operations need:

1. hands free, high capacity, highly reliable, on-line transaction data acquisition;
2. integrated solution that can record a comprehensive set of animal event types;
3. a network of distributed headless appliances acting as:
 - a. content event processors,
 - b. network sensor gateways,
 - c. network actuator controllers;
4. The ability of network components to create:
 - a. any of the standard Tracker premises event types,
 - b. any premises specific GenericEvent type;
5. event recording to high capacity enterprise data stores;
6. off line analytical processing (OLAP) of collected data with Business Intelligence Toolkits

The iRouter Premises Common Producers provides functionality to:

1. process measurements, positions, and generic events by combining Animal EIDs, Triggering states, with raw Measurements and Positions to produce Position and Measurement products that are identified with an animal EID;
2. process all the standard Tracker Premises events by combining Animal EIDs, Triggering states and Animal templates to produce Animal products.

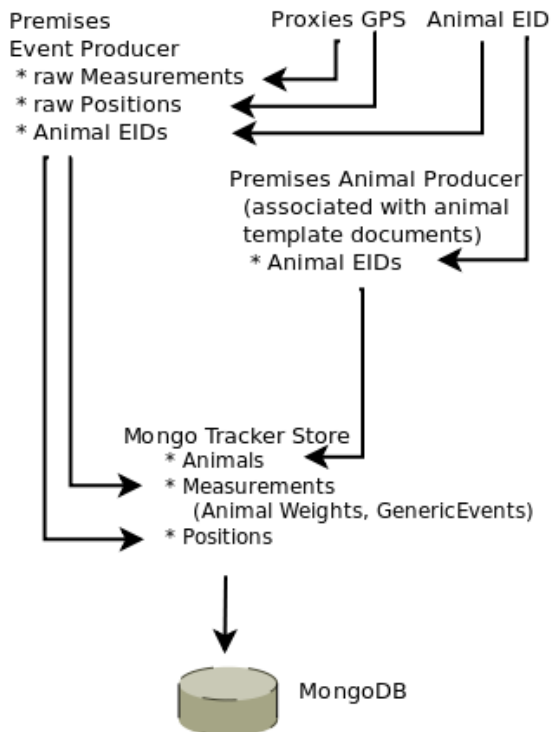
Note

See the iRouter User Guide [<http://tracker.verticon.com/Tracker/documentation/iRouter.pdf>] for details about the iRouter.

The iRouter Premises Common Producers feature contains two iRouter Component Factories.

Table 17.1. Premises Common Component Factories

| Component Factory Name | Description | Consumes | Interface input | Produces | Interface output |
|--------------------------|-------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------|-----------------|-------------------------------------------------------------------------------------|------------------|
| Premises Event Producer | Transforms raw Measurement and Position envelopes into Measurement and Position envelopes identified with an EID. | Consumes Product Envelopes that are State, Long, Measurement and Position | | Produces Envelopes that are Position, Measurement-WeighIn, Measurement-GenericEvent | |
| Premises Animal Producer | Uses an Animal template containing one or more standard events and an EID to act as a trigger | Consumes Product Envelopes that are State and Long | | Produces an Animal product containing all the events in the template | |

Figure 17.1. Premises Event Producer and Premises Animal Producer Example

Premises Event Producer

com.verticon.tracker.irouter.premises.event

Premises Event Producer is a factory for creating iRouter composite Consumer and Producer services that consume unidentified Measurement and Position envelopes and animal EID envelopes and produce Measurement and Position envelopes *identified* with an EID.

The service consumes product envelopes containing:

1. Long from EID producers
2. Position from GPS producers
3. GenericEvent from proxies
4. WeighIn from proxies
5. State as forwarding triggers

The service produces envelopes that are identified with an Animal EID and containing one of the following Values:

1. Position
2. Measurement with Envelope scope WeighIn
3. Measurement with Envelope scope associated with the OCD name of an GenericEvent

Premises Animal Producer

com.verticon.tracker.irouter.premises.animal

Premises Animal Producer is a factory for creating iRouter composite Consumer and Producer services that will produce one or more Premises events. These premises events are contained in a referenced Animal template. An EID and optional State can act as trigger condition.

The service consumes product envelopes containing:

1. Long from EID producers
2. State as forwarding triggers

The service produces envelopes containing:

1. an Animal that is identified with an Animal EID
2. where the Animals events are copied from the Animal template and timestamped with the time of trigger.

Chapter 18. MongoDB Tracker Store and Location Service

Overview

MongoDB¹ TrackerStore and Location Service² is a new generation *Big Data* storage solution for livestock market exchanges, livestock producers, livestock infrastructure providers and government animal health regulators who require:

- a highly scalable, reliable and easy to maintain information repository for recording high volumes of animal life and history event information³;
- a facility to register and record Livestock Premises geolocation details;
- the functionality for tracing animal movements within and between livestock premises;
- stored information accessible by state of the art business intelligence analysis and reporting tools;
- multi-user access controls to segregate information;
- support for iRouter client background services and appliances;
- an information exchange facility for sharing information between collaborating business partners.

Provisioned Features

To add MongoDB Tracker Store and Location Service to the Tracker provision the following features:

- MongoDB Connection Drivers *com.mongodb* feature.

MongoDB Connection Drivers *com.mongodb* feature. These are low level client drivers for connecting to one or more, local or remote MongoDB servers. See *MongoDB Java Language Center* [<http://www.mongodb.org/display/DOCS/Java+Language+Center>] for more details.

- MongoEMF Runtime *com.verticon.mongo.emf.runtime* feature.

MongoEMF sits one level up from the client drivers. MongoEMF is a persistence framework that allows information from Premises documents to be saved to MongoDB.⁴

- Tracker MongoDB Store *com.verticon.tracker.store.mongo* feature.

¹ *MongoDB* [<http://www.mongodb.org/>] (from "humongous") is a scalable, high-performance, open source NoSQL database.

² MongoDB TrackerStore and Location Service is an implementation of two Premises Editor services:

- *TrackerStore*
- *Location*.

See the Services section of the Premises Editor documentation for a detailed description of the TrackerStore and Location Service functionality.

³ MongoDB has the ability to scale data storage to 100s of TERA BYTES and the capabilities for creating high degrees of redundancy.

⁴ Both the client drivers and the MongoEMF framework are invisible to end users.

See *MongoEMF Project Wiki* [<https://github.com/BryanHunt/mongo-emf/wiki>] for more details.

Tracker MongoDB Store provides the core user functionality by implementing the Tracker Store and Location Services.⁵ As well as providing :

- Administration facility for controlling user access to information when creating information exchanges with multiple users recording and retrieving information from multiple premises.
- Administration facility for registering Livestock Premises;
- User recording and retrieval of Livestock Premises, Animal and Animal Event information;
- Desktop client support for connecting to one or multiple remote or local MongoDB servers;
- Traceability services for locating animals and analyzing movements within and between premises.
- iRouter client support
- Mongo EMF ODA Driver *com.verticon.mongo.emf.oda* feature.

Mongo EMF ODA Driver provides Eclipse Business Intelligence Reporting Tool BIRT access to EMF documents⁶ stored in MongoDB.

- Mongo EMF ODA User Interface *com.verticon.mongo.emf.oda* feature.

Mongo EMF ODA provides the graphical user interfaces which allow business analysts using Eclipse Business Intelligence Reporting Tool BIRT designer tools the graphical support to build report *datasources* and *datasets* [http://wiki.eclipse.org/BIRT/FAQ/Data_Access#Q:_What_is_a_.22data_source.22_and_a_.22data_set.22.3F] that access EMF documents stored in MongoDB.

- Mongo EMF JSON Query *com.verticon.mongo.emf.json.query* feature. This feature provides a familiar query language for use by BIRT MongoEMF report designers that is similar to the query command⁷ syntax used in the MongoDB command line shell.

Server Installation and Administration

MongoDB TrackerStore is a client server technology where a host server running MongoDB⁸ and configured with a TrackerStore *Admin* configuration can be connected to by one or more Tracker Desktop users.⁹

Administration Tasks

Once MongoDB is installed on the server, the server will have to be loaded with an *Admin* configuration document. This admin configuration specifies a user¹⁰ that will act as an *Administrator*, which users are able to publish premises resources¹¹ and what resources users can publish and retrieve from the MonboDB.

The Administrator

1. creates an *Admin* configuration document which specifies:

⁵See the Services section in the Premises Editor documentation.

⁶In the case of TrackerStore the EMF documents would be Premises and Admin documents.

⁷See *MongoDB Querying* [http://www.mongodb.org/display/DOCS/Querying] for more details.

⁸For instructions on installing MongoDB on a host server see the *MongoDB documentation* [http://www.mongodb.org/display/DOCS/Quickstart].

⁹Client side Tracker Desktops must be provisioned with MongoDB TrackerStore features.

¹⁰A computer user is referred to as a Member in the Admin document.

¹¹A Premises is called a Resource in the Admin document. A Resource is referenced to a premises by an URI (Universal Resource Identifier).

- a. members, by the user name and the computer name they will access TrackerStore from;
 - b. resources, by the uri of the Premises;
 - c. which resources members can publish;
 - d. which resources members can retrieve;
 - e. the member that is the administrator
2. loads a *Admin* configuration document into TrackerStore;¹²
 3. register premises.¹³

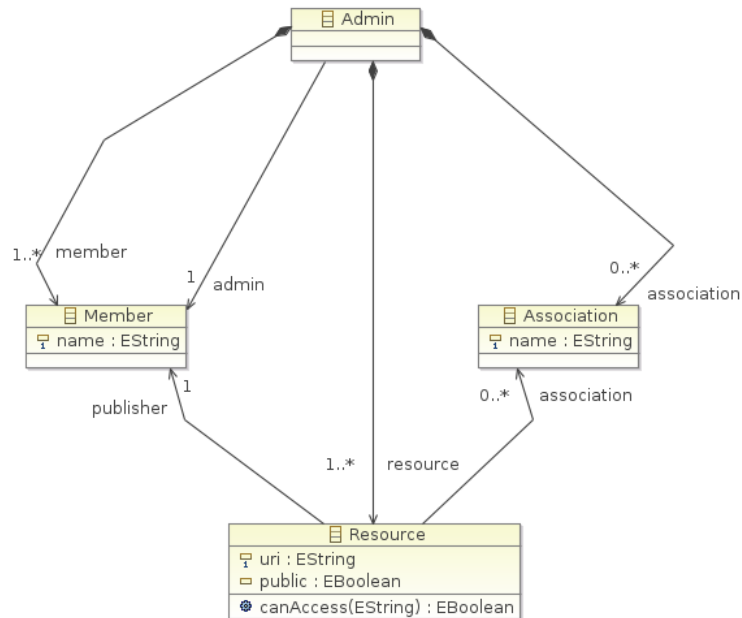
Admin Configuration Document

All Admin configuration documents structurally conform to the *Admin Data Model*. In the Admin Data Model, a single *Admin* element contains one or more *Member* elements, one or more *Resource* elements and zero or more *Association* elements. The *Admin* element specifies a *Member* as an Administrator. The *Resource* element specifies

1. a *Member* as the publisher of the resource
2. an *Association* if the *Resource* is associated with resources from other premises.

Elements of the Admin Data Model are shown in Figure 18.1, “Admin Data Model Elements”.

Figure 18.1. Admin Data Model Elements



- a. The elements of the Admin Data Model are depicted as rectangles in the figure.
- b. The notation within the elements are the attributes of the element.

¹²When the server is first installed and no configuration is loaded, any user can load an Admin document to configure the access rules.

¹³See the Premises Editor documentation, *Premises Action Commands* section for details on how an Administrator registers premises.

c. The notation between the elements are the relationships of the elements.

The following table offers a text explanation of the primary element relationships.

Table 18.1. Elements of the Admin Model

| Element Name | Relationship to other Elements |
|--------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Admin | There is one Admin element in an Admin document. The Admin element is the root element of the document. All other elements are children elements of the Admin element. An Admin element contains one or more Members, one or more Resources and zero or more Associations. The Admin element references one Member as an Admin (administrator). The Admin Editor and the Properties View are used for creating and editing Admin documents. |
| Member | The Member element represents a user on a computer. The Admin element must have one or more Member elements. Each Member has a name attribute to designate the user name and the name of the computer the user will use to access MongoDB. The name attribute takes the form <i>userName@hostName</i> . |
| Association | The Association element provides an option for creating named sets of two or more resources. The Admin element may contain zero or more Associations. The Association element has a required name attribute. |
| Resource | The Resource element represents information that can be published ^a on the MongoDB TrackerStore. A Resource must reference a Member to identify the publisher and may reference an Association to indicate it is part of an association. The Resource has an uri attribute for identification and may be designated as public. When a resource is public, it can be retrieved by any TrackerStore user. For a non-public resource that is included in an association the publishers of any of the resources in the association can retrieve it. For resources that do not reference an association only the member publisher can retrieve it. |

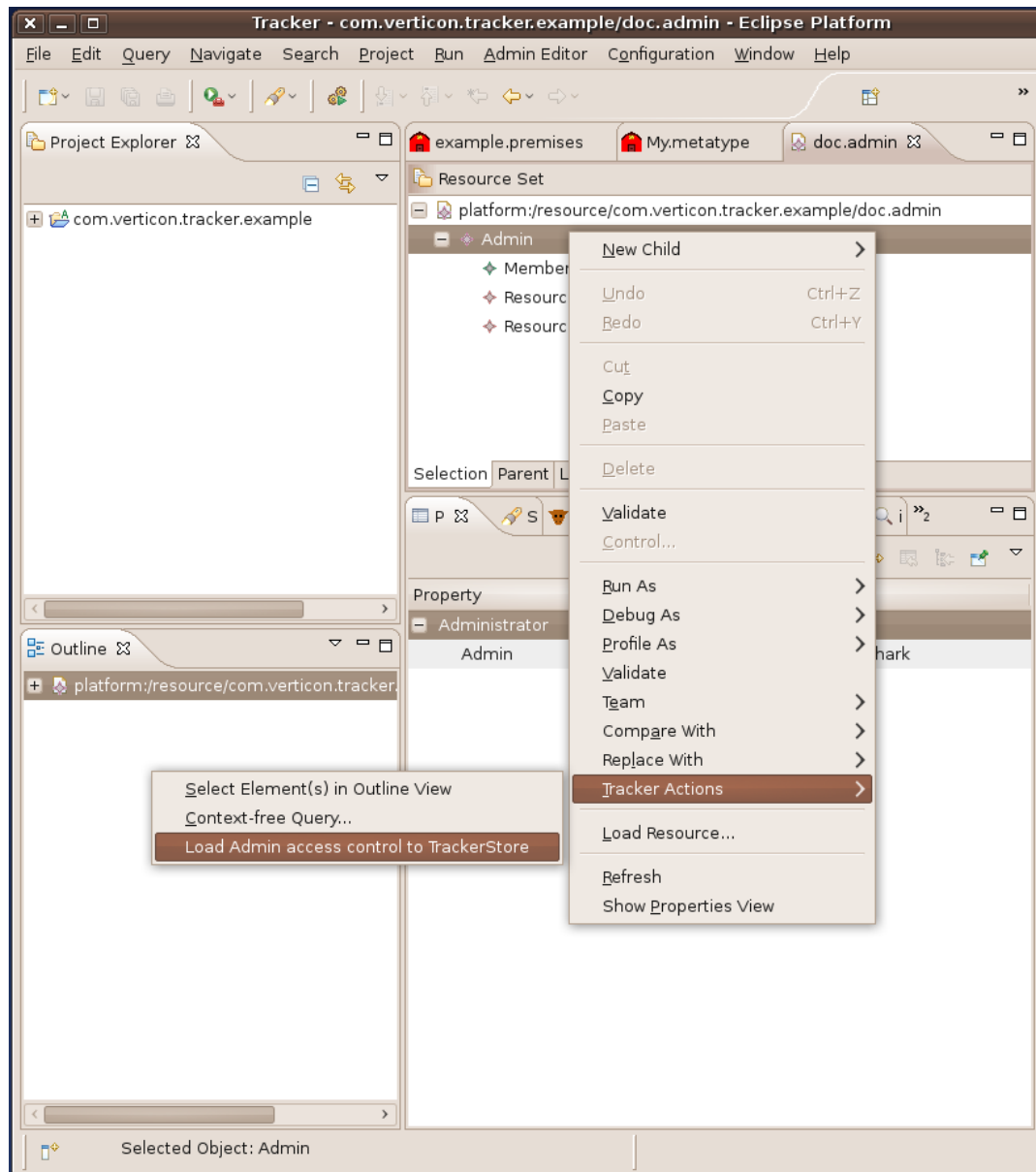
^aInitially only premises documents and contained animal information can be published.

Loading Admin Configuration

On the Admin element of the Admin document, the Admin Editor presents an action command to load the Admin document into TrackerStore.¹⁴ This action may be accessed as popup menus from the Selection page of the Admin Editor or the document tree in the Outline View tree.¹⁵

¹⁴This command is added to the base set of actions commands on the Admin element only when one or more MongoDB Tracker Store connections are configured.

Figure 18.2. Loading an Admin document from Admin Editor Selection Page



¹⁵ Right click on the element and the popup main menu will be shown. From there the sub menu 'Tracker Actions' will contain the appropriate action command.

Table 18.2. Admin Editor Document Element Action Commands

| Element | Command Name | Permission Level | Description | Comment |
|---------|--------------------------------------|-----------------------------------|-----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| Admin | Load Admin access control to MongoDB | Server Administrator ^a | Loads (stores) the Admin access controls to MongoDB | When a MongoDB server is installed the first user to load an Admin document will determine the access controls for the server. |

^aAccess controls do not exist when an Admin access control document is not loaded in MongoDB.

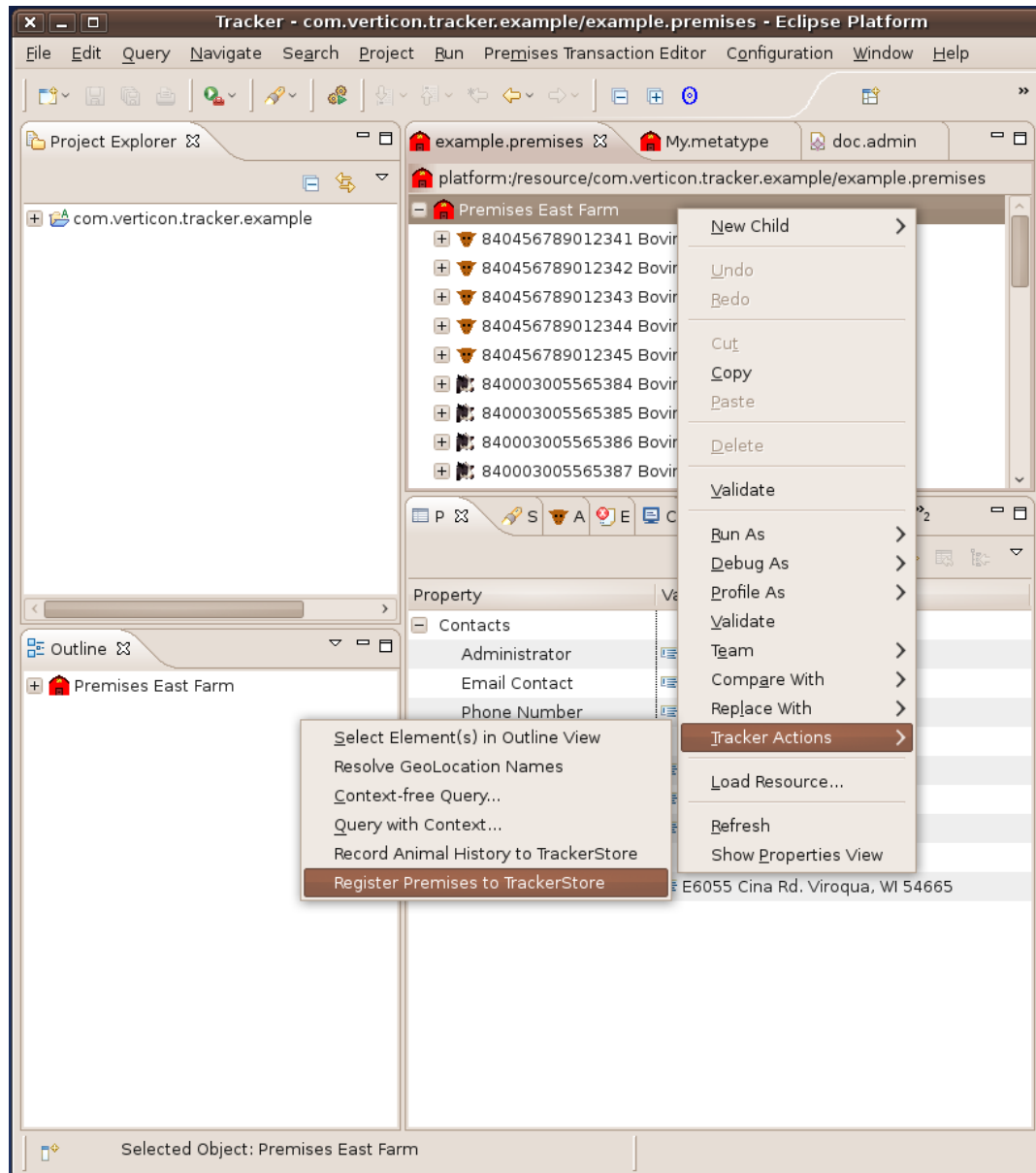
Registering a Premises

On the Premises element of the Premises document, the Premises Editor presents an action command to register the Premises to TrackerStore.¹⁶ This action may be accessed as popup menus from the Selection page of the Premises Editor or the document tree in the Outline View tree.¹⁷

¹⁶This command is added to the base set of actions commands on the Premises element only when one or more Tracker Store connections are configured and when the user is the *Administrator* for the TrackerStore. Registration of Premises is assigned to the administrator so that the premises details can be validated prior to registration.

¹⁷ Right click on the element and the popup main menu will be shown. From there the sub menu 'Tracker Actions' will contain the appropriate action command.

Figure 18.3. Register a Premises from the Premises Editor Selection Page



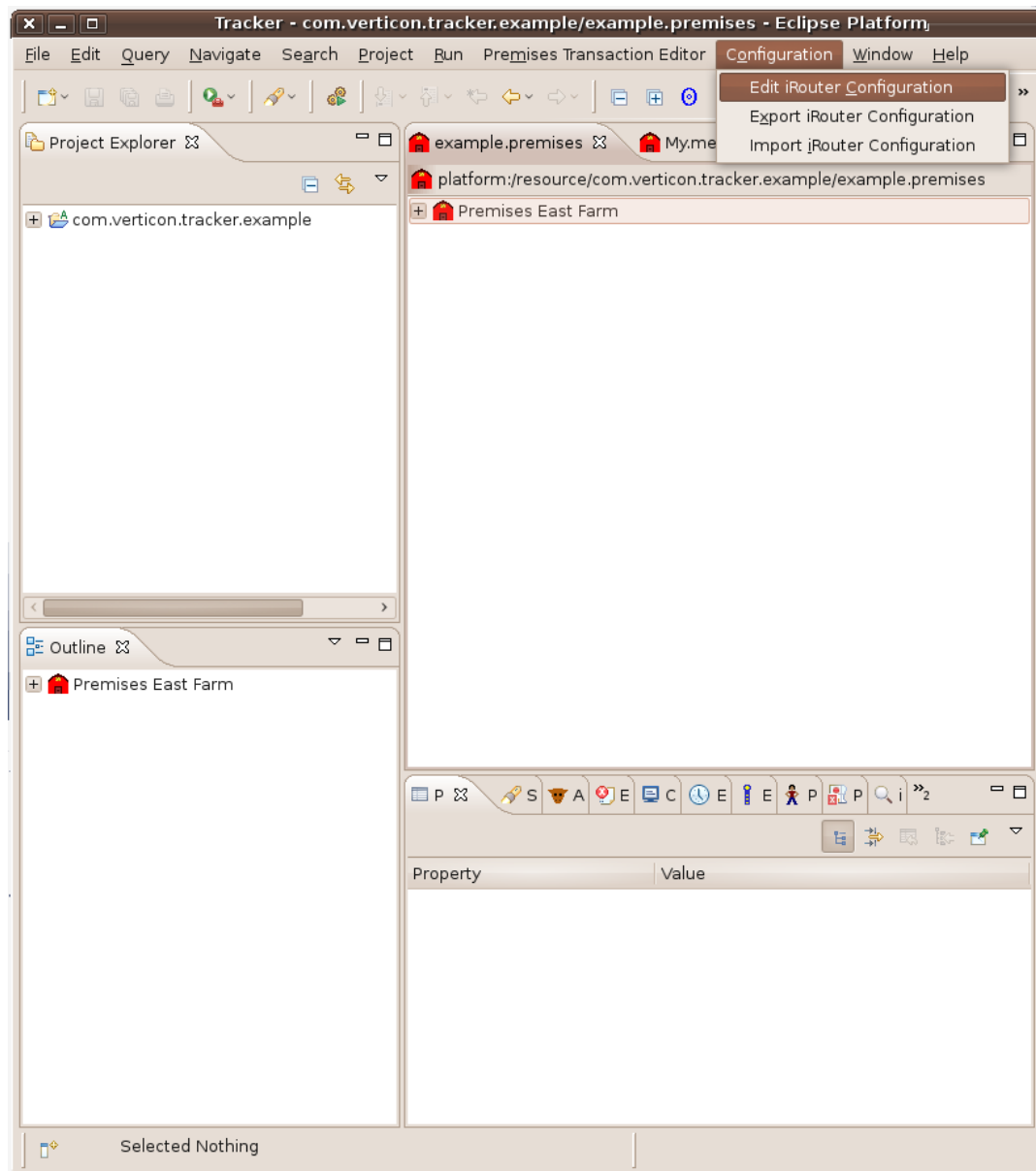
Configuring MongoDB iRouter Services

After provisioning the Tracker Desktop with the MongoDB features, the user must create and configure a MongoDB service to communicate with a MongoDB host server.¹⁸

To create a MongoDB TrackerStore service, from the menu bar select *Configuration* and *Edit iRouter Configuration*.

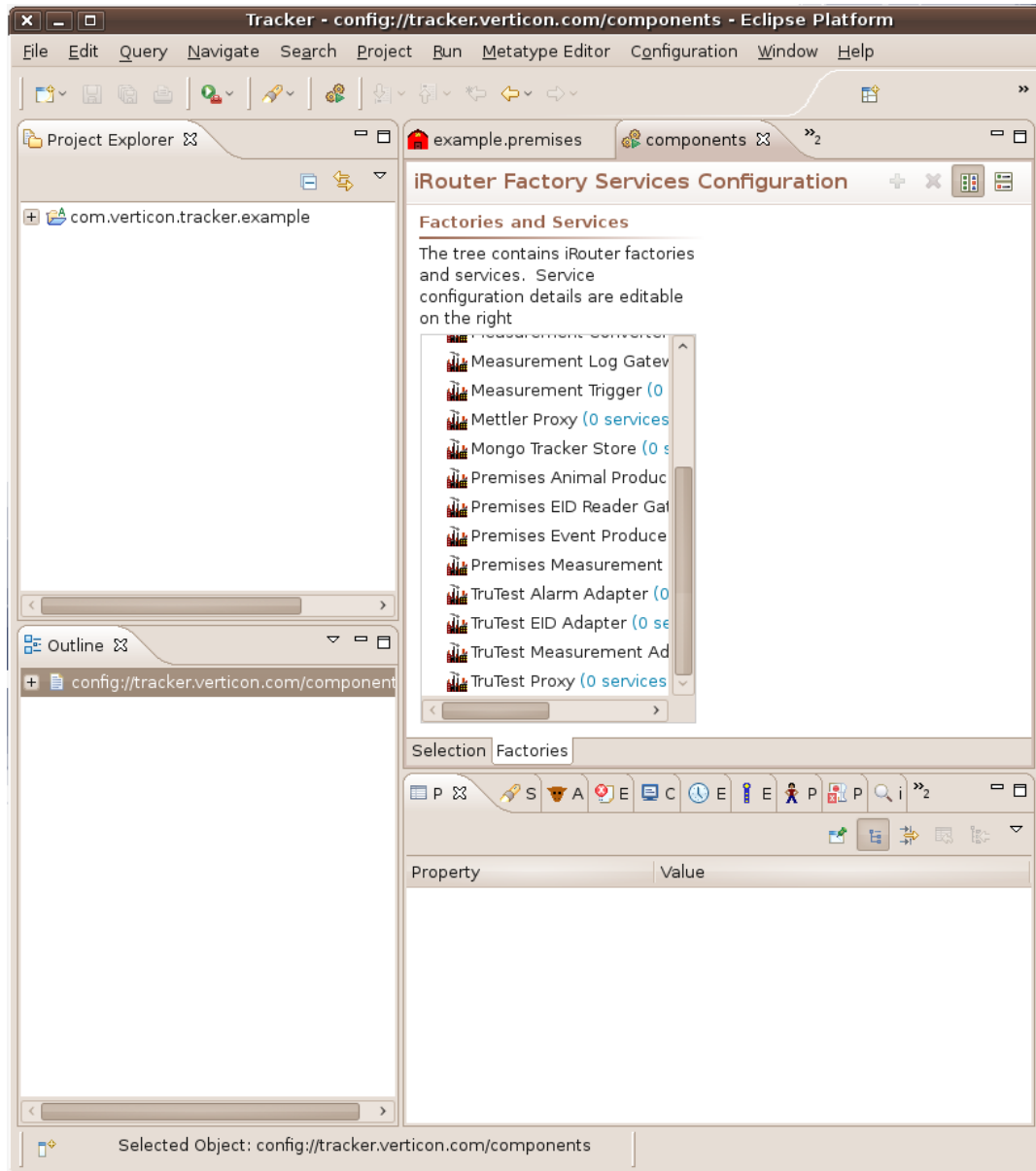
¹⁸There can be multiple services configured to connect to multiple MongoDB servers.

Figure 18.4. Edit an iRouter Configuration



The iRouter Configuration Editor will be displayed. It will have two tabs at the bottom of the editor window *Selection* and *Factories*. Select the *Factories* tab.

Figure 18.5. iRouter Editor Factories



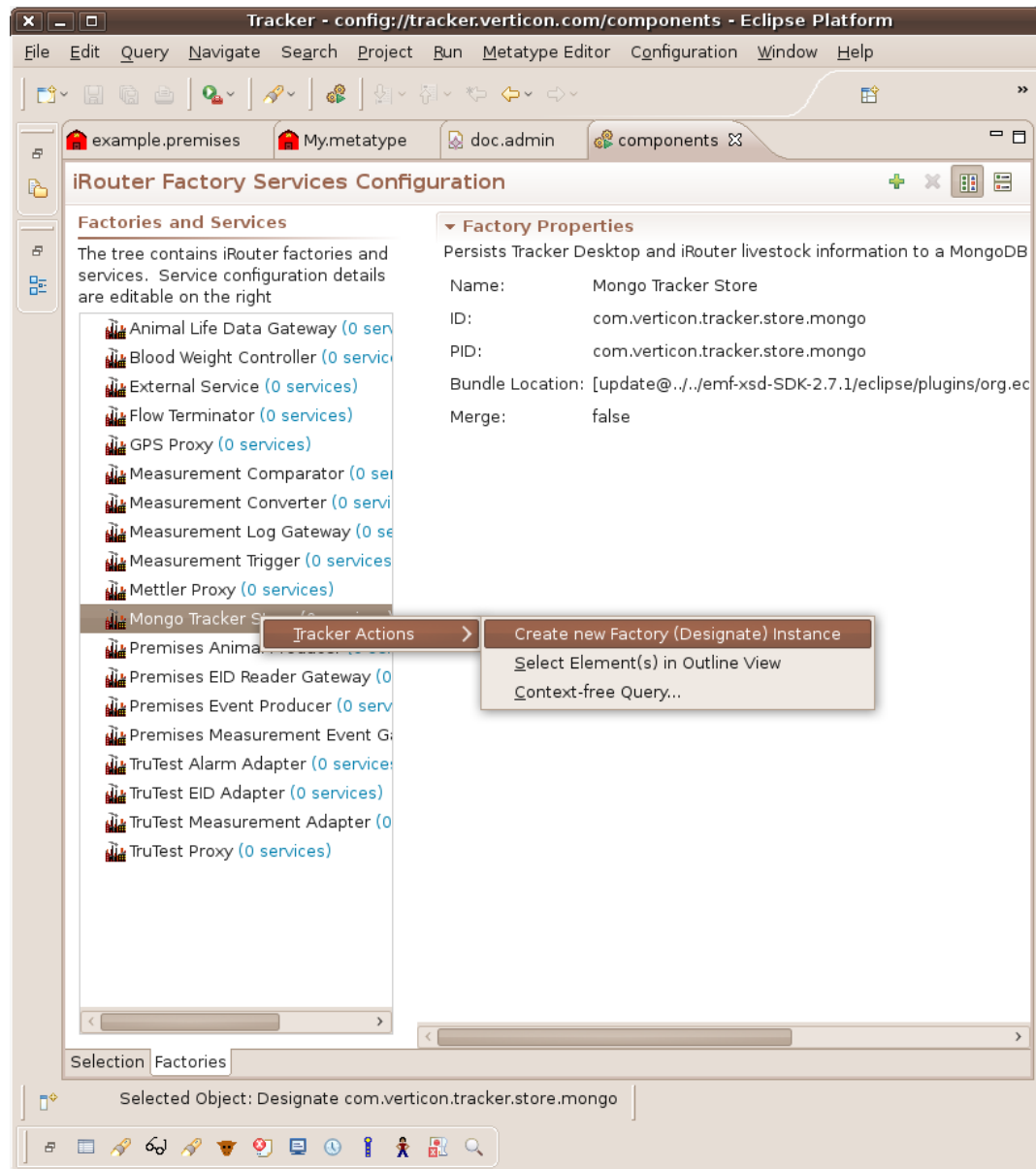
The list box in the Factories tab of the iRouter editor will contain iRouter Factories and Services.

Note

They are called factories because they are used to create a service instance.¹⁹ The Mongo Tracker Store factory will create Mongo Tracker Store services. Select the *Mongo Tracker Store* factory, right click to bring up the popup menu and select the *Create new Factory (Designate) Instance*.

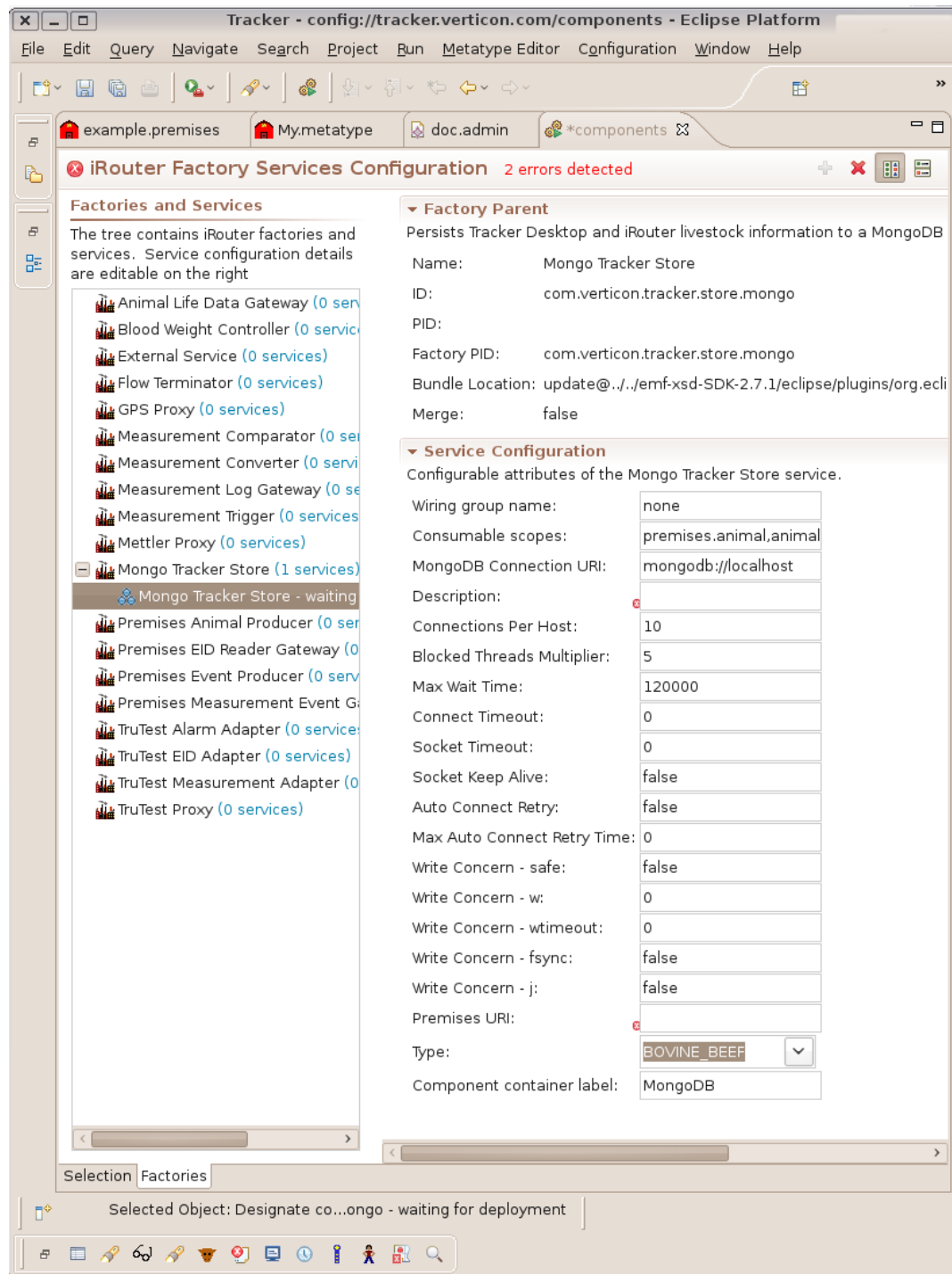
¹⁹Service instances are also called a Factory designate instances.

Figure 18.6. Creating a New Mongo TrackerStore Service



The factory will now show that the number of service instance was incremented and the new service will appear underneath the factory in the list. Select the new service and the configuration form editor will appear in the right part of the editor window.

Figure 18.7. Configuring a TrackerStore Service



A Mongo TrackerStore service has several configurable attributes because it is not only one service it is a composite of services:

1. TrackerStore Service
2. Location Service

3. iRouter Consumer Service²⁰

Configuration may look complicated, but to get a basic services for TrackerStore and Location only requires setting the following three attributes:

1. *MongoDB Connection URI*

The connection URI takes the form: `mongodb://[username:password@]host1[:port1][,host2[:port2],...[,hostN[:portN]]][/[database]]`. The default `mongodb://localhost` connects to a mongodb server running on the local host.

2. *Description*

Use any language here to describe the connection.

3. *Premises URI*

This is the URI of the premises. It has to match the premises the user will be publishing.
Example: `urn:pin:H89234X`

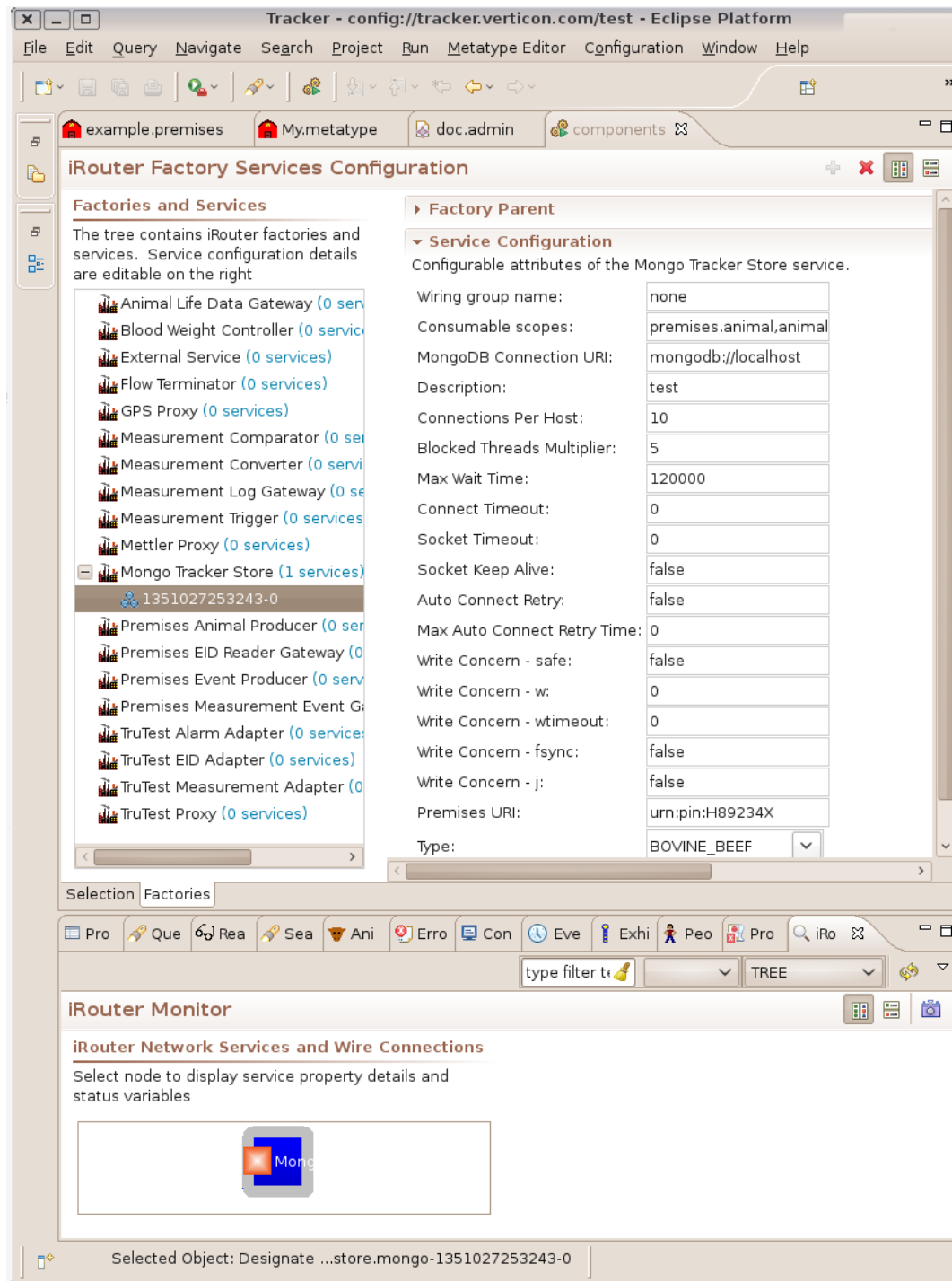
Warning

Leave all other attributes as default settings, unless you know what your doing!
Once these three attributes are set the configuration can be saved to start the service.²¹

²⁰See iRouter Consumer Service Section for details on configuring a Mongo TrackerStore instance as an iRouter consumer service.

²¹Stopping and restarting the Tracker Desktop will automatically restart the service. To delete the service, edit the configuration and delete the service instance and save the configuration.

Figure 18.8. TrackerStore Service Running and Shown in the iRouter Monitor View



iRouter Consumer Service

The Mongo TrackerStore is also an iRouter Consumer²² that can be configured within a iRouter network to consume products from iRouter Producers. The Mongo TrackerStore iRouter Consumer can be used

²²Technically a *org.osgi.service.wireadmin.Consumer* service.

by iRouter managed sensor and measurement networks for real-time recording of Tracker Premises transactions.

Mongo TrackerStore Consumer Service:

1. can be configured and run in Tracker Desktop BI Toolkit for hands free saving of animals, tags and events that originate from iRouter networks of sensor and
2. can be configured and run in iRouter headless appliances for automatic recording of tag events from one or more fixed sensors
3. can be configured and run as multiple services instances to connect to more than one MongoDB stores
4. supports event updates to existing animals and the runtime creation of new animals
5. leverages the power of the iRouter infrastructure for creating complex sensor and actuator based solutions.
6. can support recording of events from one or more tag readers or virtual tag producers.

The Mongo TrackerStore consumes:

1. Tags
 - a. Tags can contain one or more events;
 - b. Saving the tag in a new animal created from the default animal type²³ if it is a new tag id;
 - c. Saving the latest events if the tag is already persisted;
 - d. Saving the tag in an existing animal if this tag contains a replacedTag event.
2. Animals²⁴

Saving the first tag in the animal as described above in saving tags.
3. Envelopes containing GenericEvents²⁵ where the envelope:
 - a. id must be a Long or String object as that will be used for the Tag id,
 - b. envelope scope is the OCD,
 - c. envelope contents are used to create the event.

²³The default Animal type is set in the *Type* attribute of the Mongo TrackerStore Service Configuration.

²⁴In the iRouter Example Section the *Premises Animal Producer* is an example of a Producer service that produces Animal products.

²⁵In the iRouter Example Section the *Premises Event Producer* is an example of a Producer service that produces Envelope Measurement products. (Premises Event Producer can produce Positions as well, but in the example it is configured to produce only Measurements).

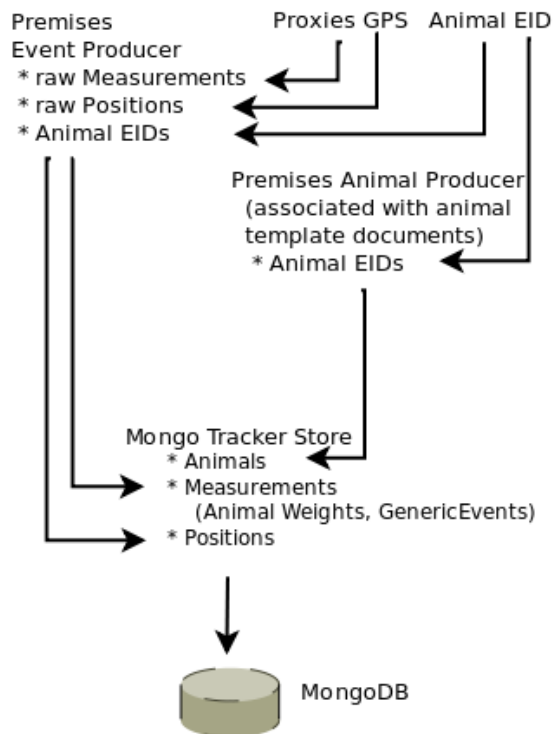
Table 18.3. Mongo Tracker Store iRouter Consumer

| Component Factory Name | Description | Consumes | Interface input | Produces | Interface output |
|---------------------------|---------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|-----------------|----------|-------------------------------------------|
| Mongo Tracker Store | Stores standard and generic animal events to the TrackerStore | Consumes Product Envelopes that are Animal, Tag, Measurement and Position | | | Stores products to the TrackerStore |

iRouter Example

Goal: Demonstrate an iRouter component configuration that highlights the functionality of the Tracker Mongo Store.

Figure 18.9. TrackerStore Service iRouter Network Product Flows



1. Components used

a. TruTest Proxy

- This will serve at the top of the event production chain to produce events associated with a TruTest scale head indicator
- The scale will produce events as the weights of the animals are changed, these weight measurements will be sent in the *raw.animal.weight* scope
- When an animal EID is detected on the scale head that eid will be produced in the *animal.eid* scope

- When the operator presses the Enter Button that event will be produced in the *transaction.state* scope

b. Premises Event Producer

- This component will consume all products sent by the TruTest Proxy and on the detection of a *transaction.state* indicating that the operator pressed the Enter Button, this component will produce the last measurement received and identify it with the animal eid.

c. Premises Animal Producer

- This component also sits towards the top of the event production chain, but consumes Animal EIDs on the *animal.eid* scope and Enter Button presses on the *transaction.state* scope that have originated on the TruTest Proxy. Associated with this component is a *URI* referencing an Tracker Premises Animal template in an external document. Production cycle of this component starts when a Enter Button press is detected. Then the Animal Template is copied, given the EID from the TruTest, and sent to consumers.

d. Tracker Mongo Store consumes products from the Premises Event Producer and the Premises Animal Producer and creates these on the MongoDB.

2. Startup and Configuration

a. MongoDB must be installed and running on the local host

b. Startup Tracker

c. From the *Configuration* menu, invoke *Import iRouter Configuration*

d. Import that configuration *autotype.metatype* document in this project

3. TruTest simulator

a. Download and install the Tracker simulator product

b. In the installation directory run the simulator executable

c. In a web browser go to Simulated Devices [<http://localhost:8081/client/devices>] and start the TruTest simulator

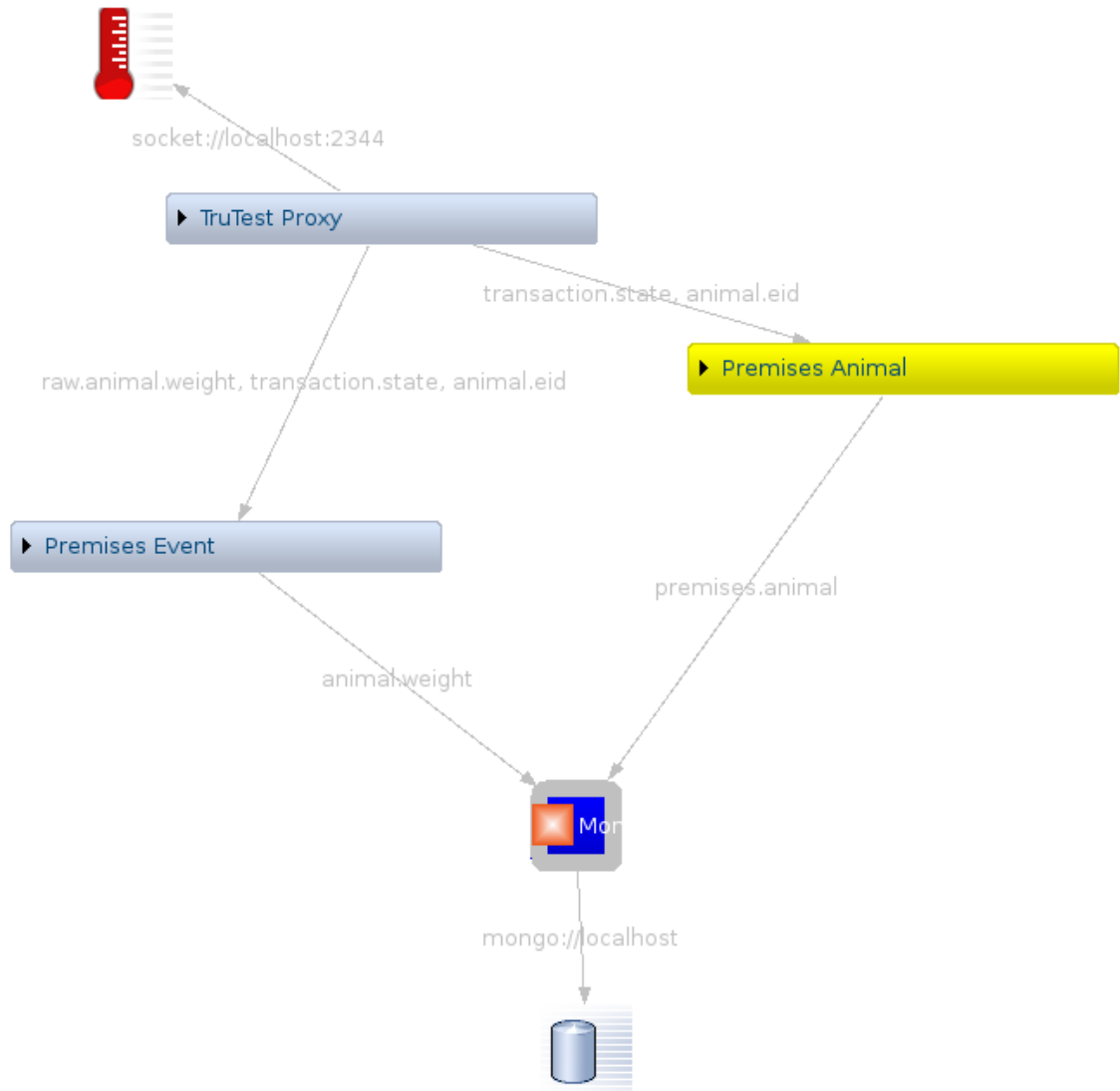
d. Select a TruTest simulator

e. When the TruTest simulator comes up the *Connection* should be Y

f. Change the EID, the weight and press the record button to simulate a transaction

iRouter Monitor View of the Demonstration

In the iRouter Monitor View you should see the components displayed something like:



Appendix A. GNU Free Documentation License

Copyright (C) 2000, 2001, 2002 Free Software Foundation, Inc. 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA. Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover,

and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of

the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright (C) YEAR YOUR NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Appendix B. Eclipse Foundation Software User Agreement

Eclipse Foundation Software User Agreement

February 1, 2011

Usage Of Content

THE ECLIPSE FOUNDATION MAKES AVAILABLE SOFTWARE, DOCUMENTATION, INFORMATION AND/OR OTHER MATERIALS FOR OPEN SOURCE PROJECTS (COLLECTIVELY "CONTENT"). USE OF THE CONTENT IS GOVERNED BY THE TERMS AND CONDITIONS OF THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. BY USING THE CONTENT, YOU AGREE THAT YOUR USE OF THE CONTENT IS GOVERNED BY THIS AGREEMENT AND/OR THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW. IF YOU DO NOT AGREE TO THE TERMS AND CONDITIONS OF THIS AGREEMENT AND THE TERMS AND CONDITIONS OF ANY APPLICABLE LICENSE AGREEMENTS OR NOTICES INDICATED OR REFERENCED BELOW, THEN YOU MAY NOT USE THE CONTENT.

Applicable Licenses

Unless otherwise indicated, all Content made available by the Eclipse Foundation is provided to you under the terms and conditions of the Eclipse Public License Version 1.0 ("EPL"). A copy of the EPL is provided with this Content and is also available at <http://www.eclipse.org/legal/epl-v10.html>. For purposes of the EPL, "Program" will mean the Content.

Content includes, but is not limited to, source code, object code, documentation and other files maintained in the Eclipse Foundation source code repository ("Repository") in software modules ("Modules") and made available as downloadable archives ("Downloads").

- Content may be structured and packaged into modules to facilitate delivering, extending, and upgrading the Content. Typical modules may include plug-ins ("Plug-ins"), plug-in fragments ("Fragments"), and features ("Features").
- Each Plug-in or Fragment may be packaged as a sub-directory or JAR (Java™ ARchive) in a directory named "plugins".
- A Feature is a bundle of one or more Plug-ins and/or Fragments and associated material. Each Feature may be packaged as a sub-directory in a directory named "features". Within a Feature, files named "feature.xml" may contain a list of the names and version numbers of the Plug-ins and/or Fragments associated with that Feature.
- Features may also include other Features ("Included Features"). Within a Feature, files named "feature.xml" may contain a list of the names and version numbers of Included Features.

The terms and conditions governing Plug-ins and Fragments should be contained in files named "about.html" ("Abouts"). The terms and conditions governing Features and Included Features should be contained in files named "license.html" ("Feature Licenses"). Abouts and Feature Licenses may be located in any directory of a Download or Module including, but not limited to the following locations:

- The top-level (root) directory
- Plug-in and Fragment directories
- Inside Plug-ins and Fragments packaged as JARs
- Sub-directories of the directory named “src” of certain Plug-ins
- Feature directories

Note: if a Feature made available by the Eclipse Foundation is installed using the Provisioning Technology (as defined below), you must agree to a license (“Feature Update License”) during the installation process. If the Feature contains Included Features, the Feature Update License should either provide you with the terms and conditions governing the Included Features or inform you where you can locate them. Feature Update Licenses may be found in the “license” property of files named “feature.properties” found within a Feature. Such Abouts, Feature Licenses, and Feature Update Licenses contain the terms and conditions (or references to such terms and conditions) that govern your use of the associated Content in that directory.

THE ABOUTS, FEATURE LICENSES, AND FEATURE UPDATE LICENSES MAY REFER TO THE EPL OR OTHER LICENSE AGREEMENTS, NOTICES OR TERMS AND CONDITIONS. SOME OF THESE OTHER LICENSE AGREEMENTS MAY INCLUDE (BUT ARE NOT LIMITED TO):

- Eclipse Distribution License Version 1.0 (available at <http://www.eclipse.org/legal/epl-v10.html>)
- Common Public License Version 1.0 (available at <http://www.eclipse.org/legal/cpl-v10.html>)
- Apache Software License 1.1 (available at <http://www.apache.org/licenses/LICENSE>)
- Apache Software License 2.0 (available at <http://www.apache.org/licenses/LICENSE-2.0>)
- Metro Link Public License 1.00 (available at <http://www.opengroup.org/openmotif/supporters/metrolink/license.html>)
- Mozilla Public License Version 1.1 (available at <http://www.mozilla.org/MPL/MPL-1.1.html>)

IT IS YOUR OBLIGATION TO READ AND ACCEPT ALL SUCH TERMS AND CONDITIONS PRIOR TO USE OF THE CONTENT. If no About, Feature License, or Feature Update License is provided, please contact the Eclipse Foundation to determine what terms and conditions govern that particular Content.

Use of Provisioning Technology

The Eclipse Foundation makes available provisioning software, examples of which include, but are not limited to, p2 and the Eclipse Update Manager (“Provisioning Technology”) for the purpose of allowing users to install software, documentation, information and/or other materials (collectively “Installable Software”). This capability is provided with the intent of allowing such users to install, extend and update Eclipse-based products. Information about packaging Installable Software is available at http://eclipse.org/equinox/p2/repository_packaging.html (“Specification”).

You may use Provisioning Technology to allow other parties to install Installable Software. You shall be responsible for enabling the applicable license agreements relating to the Installable Software to be presented to, and accepted by, the users of the Provisioning Technology in accordance with the Specification. By using Provisioning Technology in such a manner and making it available in accordance with the Specification, you further acknowledge your agreement to, and the acquisition of all necessary rights to permit the following:

- A series of actions may occur (“Provisioning Process”) in which a user may execute the Provisioning Technology on a machine (“Target Machine”) with the intent of installing, extending or updating the functionality of an Eclipse-based product.
- During the Provisioning Process, the Provisioning Technology may cause third party Installable Software or a portion thereof to be accessed and copied to the Target Machine.
- Pursuant to the Specification, you will provide to the user the terms and conditions that govern the use of the Installable Software (“Installable Software Agreement”) and such Installable Software Agreement shall be accessed from the Target Machine in accordance with the Specification. Such Installable Software Agreement must inform the user of the terms and conditions that govern the Installable Software and must solicit acceptance by the end user in the manner prescribed in such Installable Software Agreement. Upon such indication of agreement by the user, the provisioning Technology will complete installation of the Installable Software.

Cryptography

Content may contain encryption software. The country in which you are currently may have restrictions on the import, possession, and use, and/or re-export to another country, of encryption software. BEFORE using any encryption software, please check the country's laws, regulations and policies concerning the import, possession, or use, and re-export of encryption software, to see if this is permitted.

Java and all Java-based trademarks are trademarks of Oracle Corporation in the United States, other countries, or both.