

IBM SPSS Modeler Entity Analytics 15 User Guide



Note: Before using this information and the product it supports, read the general information under Notices on p. 66.

This edition applies to and to all subsequent releases and modifications until otherwise indicated in new editions.

Adobe product screenshot(s) reprinted with permission from Adobe Systems Incorporated.

Microsoft product screenshot(s) reprinted with permission from Microsoft Corporation.

Licensed Materials - Property of IBM

U.S. Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Preface

IBM® SPSS® Modeler is the IBM Corp. enterprise-strength data mining workbench. SPSS Modeler helps organizations to improve customer and citizen relationships through an in-depth understanding of data. Organizations use the insight gained from SPSS Modeler to retain profitable customers, identify cross-selling opportunities, attract new customers, detect fraud, reduce risk, and improve government service delivery.

SPSS Modeler's visual interface invites users to apply their specific business expertise, which leads to more powerful predictive models and shortens time-to-solution. SPSS Modeler offers many modeling techniques, such as prediction, classification, segmentation, and association detection algorithms. Once models are created, IBM® SPSS® Modeler Solution Publisher enables their delivery enterprise-wide to decision makers or to a database.

About IBM Business Analytics

IBM Business Analytics software delivers complete, consistent and accurate information that decision-makers trust to improve business performance. A comprehensive portfolio of [business intelligence](#), [predictive analytics](#), [financial performance and strategy management](#), and [analytic applications](#) provides clear, immediate and actionable insights into current performance and the ability to predict future outcomes. Combined with rich industry solutions, proven practices and professional services, organizations of every size can drive the highest productivity, confidently automate decisions and deliver better results.

As part of this portfolio, IBM SPSS Predictive Analytics software helps organizations predict future events and proactively act upon that insight to drive better business outcomes. Commercial, government and academic customers worldwide rely on IBM SPSS technology as a competitive advantage in attracting, retaining and growing customers, while reducing fraud and mitigating risk. By incorporating IBM SPSS software into their daily operations, organizations become predictive enterprises – able to direct and automate decisions to meet business goals and achieve measurable competitive advantage. For further information or to reach a representative visit <http://www.ibm.com/spss>.

Technical support

Technical support is available to maintenance customers. Customers may contact Technical Support for assistance in using IBM Corp. products or for installation help for one of the supported hardware environments. To reach Technical Support, see the IBM Corp. web site at <http://www.ibm.com/support>. Be prepared to identify yourself, your organization, and your support agreement when requesting assistance.

Contents

1	<i>Entity analytics</i>	1
	About entity analytics	1
	Entity analytics and predictive analytics	2
2	<i>Entity analytics with IBM SPSS Modeler</i>	4
	Using entity analytics with IBM SPSS Modeler	4
	Stage 1: Read the source data into SPSS Modeler	5
	Stage 2: Create the repository	6
	Stage 3: Connect SPSS Modeler to the repository	7
	Stage 4: Map the input fields to repository features	7
	Stage 5: Export data to the repository and resolve conflicts	8
	Stage 6: Analyzing the resolved identities	11
	Stage 7: Resolve new cases against the repository	11
	Stage 8: Generate alerts	13
3	<i>Entity analytics tasks</i>	14
	About the tasks	14
	Setting up an entity repository (EA Export node)	14
	The entity repository	14
	Connecting to a data source	15
	Creating the repository	16
	Mapping input fields to features (EA Export node)	19
	Displaying the field mappings (EA Export node)	21
	Configuring an entity repository	22
	Viewing the data source mappings	23
	Maintaining the repository features	24
	Adding or editing a feature	26
	Maintaining the entity types	28
	Setting the threshold for entity matching	30
	Reusing a repository configuration	31
	Saving your configuration changes	32
	Closing the configuration window	32
	Analyzing the resolved identities (Entity Analytics(EA) source node)	32
	Selecting a data source	32
	Renaming data fields	33

Setting type information for data fields	33
Adding nodes to the stream	34
Comparing new cases against the repository (Streaming EA node)	34
Mapping input fields to features (Streaming EA node)	35
Displaying the field mappings (Streaming EA node)	37
Viewing the data sources (Streaming EA node)	38
Output from the Streaming EA node	39
Using IBM SPSS Modeler Entity Analytics with other IBM SPSS products	39
Administrative tasks	39
Configuring port assignments	40
Managing administrator credentials for the repository database	41
Moving the repository to a different storage directory	41
Setting stream properties for date/time and timestamp fields	42
Adjusting the timeout settings	42
Running IBM SPSS Modeler Entity Analytics with SPSS Modeler client and SPSS Modeler Server on the same Windows system	43
Purging an entity repository	43
Deleting an entity repository	44
Deleting a repository when unable to connect to it	44

4 Entity analytics in action 46

About this example	46
The original model	46
Adding entity analytics	50
Getting the source data into the repository	51
Reading the resolved identities	52
Comparing entity analytics output with the original model	58
Summary	62

Appendices

A Scripting Properties for IBM SPSS Modeler Entity Analytics 64

Common properties	64
entityanalytics_exportnode properties	64
entityanalytics_sourcenode properties	65
entityanalytics_processnode properties	65

B Notices

66

Index

69

Entity analytics

About entity analytics

IBM® SPSS® Modeler Entity Analytics adds a completely new dimension to IBM® SPSS® Modeler predictive analytics. Whereas predictive analytics attempts to predict future behavior from past data, entity analytics focuses on improving the coherence and consistency of current data by resolving identity conflicts within the records themselves. An identity can be that of an individual, an organization, an object, or any other entity for which ambiguity might exist. Identity resolution can be vital in a number of fields, including customer relationship management, fraud detection, anti-money laundering, and national and international security.

Suppose that you have the following customer records from two different sources, and are not sure whether they refer to the same person or different people.

Source 1

Record no.: 70001
Name: Jon Smith
Address: 123 Main Street
Tax Reference: 555-00-1111
Driv. License: 0001133107
Cred. Card: 10229127

Source 2

Record no.: 9103
Name: JOHNATHAN Smith
Date of Birth: 06/17/1934
Telephone: 555-1212
Cred. Card: 10229128
Email: jls@mail.com
IP address: 9.50.18.77

There are no exact matches in the data between the two records. However, if we introduce a third source, we find some common attributes.

Source 3

Record no.: 6251
Name: Jon Smith
Telephone: 555-1212
Driv. License: 0001133107
Cred. Card: 10229132

The driving license number links the records in Source 1 and Source 3, while the telephone number links Sources 2 and 3. So we can be reasonably sure that all three sources refer to the same person.

But what if the distinction is not so easy to make? We may have very little data on which to base our decision. Consider the following two records.

Source 4

Record no.: S45286
Name: John T Smith Jr
Address: 456 Main Street
Telephone: 703-555-2000
Date of birth: 03/12/1984

Record no.: S45287
Name: John T Smith
Address: 456 Main Street
Telephone: 703-555-2000
Div. License: 009900991

Evidently this is not the same Mr Smith from the previous records—the differences are sufficient that we can rule this out. However, we do still have a problem. Two different records, from the same data source, appear to relate to the same person. Are they duplicate records? We cannot be sure unless we can find another related record giving us more information, perhaps from a different source.

Source 5

Record no.: 769582-2
Name: John T Smith Sr
Address: 456 Main Street
Telephone: 703-555-2000
Div. License: 009900991
Date of birth: 06/25/1959

This resolves the problem. The two records in Source 4 are not duplicates, but are actually a father and son with the same name, living at the same address, and using the same telephone number. On a manual system, it could take weeks of searching to find the one record that resolved the identities. With an automated entity analytics system, resolution time is dramatically reduced.

Entity analytics and predictive analytics

If all of your data consisted of a single source of records that were complete and unambiguous, it would be relatively simple for IBM® SPSS® Modeler to resolve any identity conflicts. Using only predictive analytics, you could read your data into SPSS Modeler, perform your processing and obtain reliable results.

In the real world, however, the picture is normally very different. Data is typically far from complete, frequently ambiguous, and often scattered over many different data sources, recording many different attributes with few overlapping fields. Part of the value of entity analytics lies in collecting data from all the different sources into a single, central storage area, known as a **repository**. The entity analytics system then examines the data in minute detail to resolve conflicts, adding a unique identifier to records that originate from the same person or organization.

The following table illustrates the differences between the two types of analytics.

Table 1-1
Differences between predictive and entity analytics

Characteristic	Predictive analytics	Entity analytics
Training	Based on relatively small sets and numeric ranges	Based on large sets, e.g. names, addresses
Data used for training	Subset (training partition)	All data used
Generalization	Algorithm generalizes across training data to form concise model	Data persisted in structures suitable for entity matching and relationship detection
Fraud detection	Records flagged as potentially fraudulent if they have typical characteristics of fraudulent application	Records flagged as potentially fraudulent if related to known fraudulent records, or if originating from same individuals but with different identities

Entity analytics with IBM SPSS Modeler

Using entity analytics with IBM SPSS Modeler

You realize that you may have identity problems with your data. How can IBM® SPSS® Modeler Entity Analytics help you address this? The following is a suggested procedure, though you may need to vary this to suit your particular requirements.

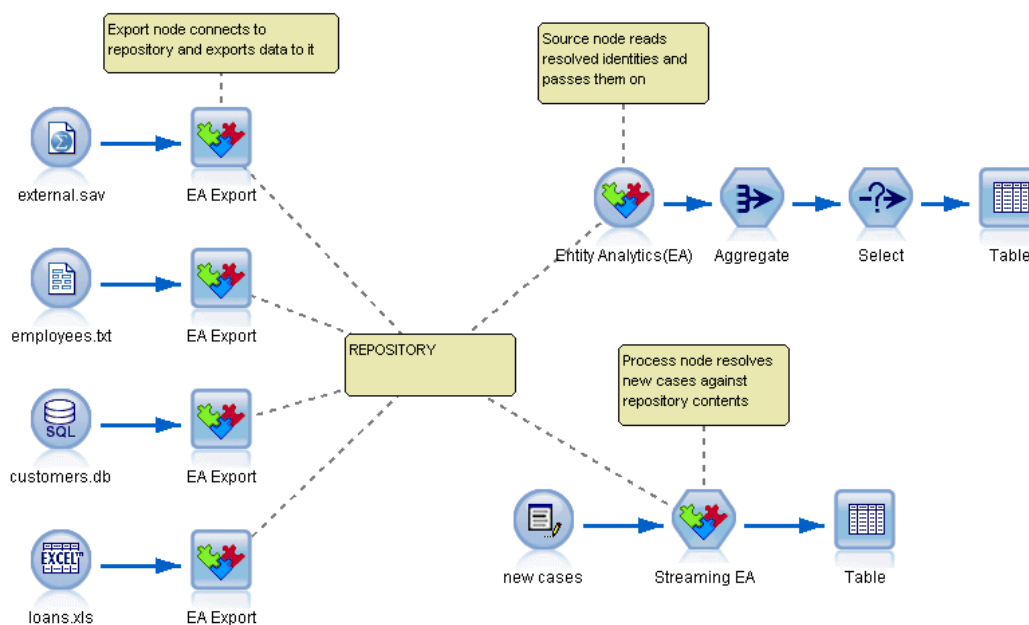
- Read the source data into IBM® SPSS® Modeler
- Create a repository ready to store the data
- Connect SPSS Modeler to the repository
- Map the data fields to repository features
- Export the data into the repository and resolve the identities
- Analyze the resolved identities
- Resolve new cases against the repository
- Generate any necessary alerts (batch or real-time)

At this point, you need to know something of how SPSS Modeler works. SPSS Modeler is a very user-friendly tool, based on the graphical representation of a stream of data flowing through a number of nodes. Each node represents a particular stage of the workflow.

SPSS Modeler itself provides a wide range of nodes, covering all the standard data mining functions. IBM SPSS Modeler Entity Analytics adds nodes for use specifically in entity analytics. These are EA Export node, the Entity Analytics(EA) source node, and the Streaming EA process node.

The following figure illustrates the process.

Figure 2-1
The entity analytics process



Stage 1: Read the source data into SPSS Modeler

Your first task is to read your data into SPSS Modeler by means of a source node, denoted in SPSS Modeler by a round icon.

Figure 2-2
IBM SPSS Modeler source node



The data can be in any format supported by SPSS Modeler, such as text files, database tables, spreadsheets, XML files, and so on, but each different format requires a corresponding SPSS Modeler source node. In the illustration, it is a Database source node.

Each data source file must have one field that uniquely identifies each record. If a data source does not have such a field, you can easily add one in SPSS Modeler. For more information, see the topic [Adding a unique record identifier](#) in Chapter 3 on p. 15.

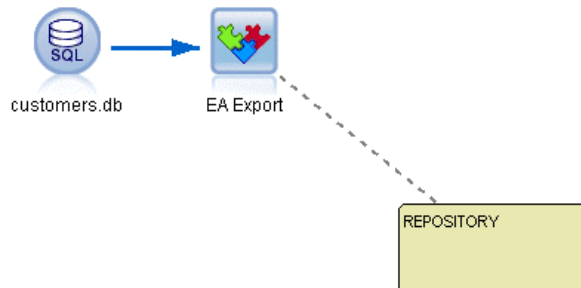
For more information, see the topic [Connecting to a data source](#) in Chapter 3 on p. 15.

Stage 2: Create the repository

The focal point of all your entity analytics efforts is the repository—the central storage area where you collect together all your data records.

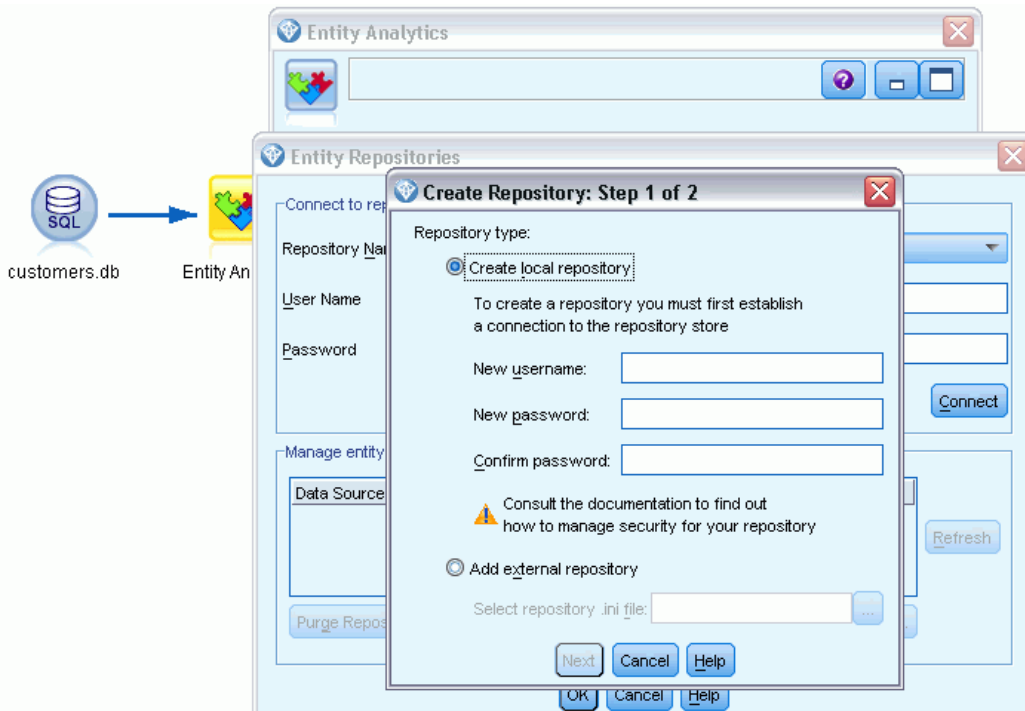
To create a repository, you start by connecting the data source to an EA Export node, represented by the square icon.

Figure 2-3
Connecting to a repository



From the export node you can create a new repository (or select an existing one), ready to receive the exported data.

Figure 2-4
Creating a repository



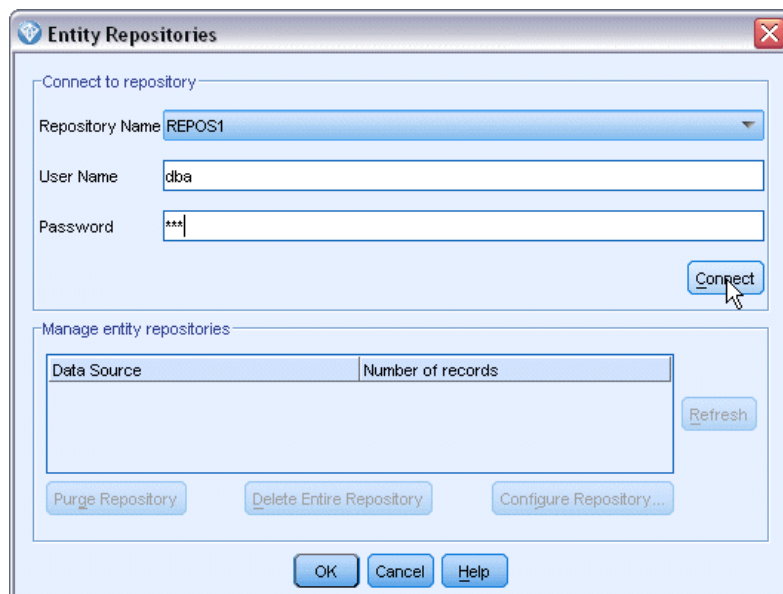
The process of creating a repository is fully described later. For more information, see the topic [Setting up an entity repository \(EA Export node\)](#) in Chapter 3 on p. 14.

When you have set up a repository, you can maintain its contents in various ways. For more information, see the topic [Configuring an entity repository](#) in Chapter 3 on p. 22.

Stage 3: Connect SPSS Modeler to the repository

Having created the repository, you then connect it to the SPSS Modeler stream.

Figure 2-5
Connecting to the repository



For more information, see the topic [Entity repository options](#) in Chapter 3 on p. 18.

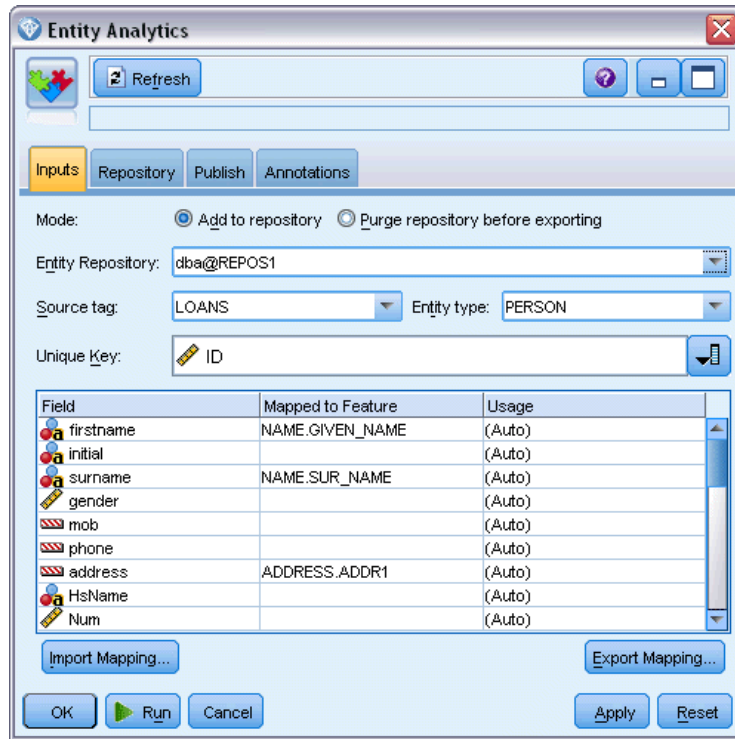
Stage 4: Map the input fields to repository features

Data sources can contain many different kinds of entity information. Some information types are common to most entity data sources, while others may be specific to a particular data source. In an entity repository, these different information types are known as **features**. The repository provides a number of features as standard, and you can also create your own features.

A repository feature is an individual information type that can be used with an entity data source. Some features (for example, First Name, Last Name, Date of Birth, and so on) may be capable of being used with many different data sources, while other features will be specific to a particular data source. A feature is typically the equivalent of a field in a data record, or a column in a database table.

When you have created a repository and connected to it, you designate one field of your input data as the **unique key** field, which is used in subsequent analysis. You also map the input data fields to their corresponding features in the repository. This is to avoid duplication in the repository in cases where different data sources use different names for fields containing the same kind of information. The EA Export node provides a mapping table where you can create the mappings.

Figure 2-6
Mapping input fields to features

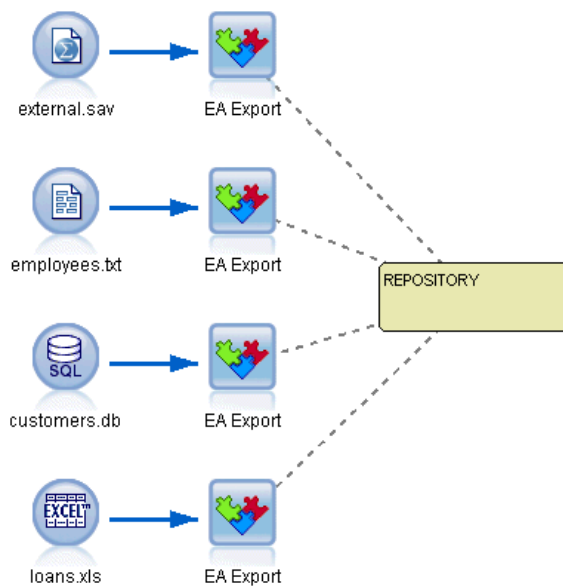


For more information, see the topic [Mapping input fields to features \(EA Export node\)](#) in Chapter 3 on p. 19.

Stage 5: Export data to the repository and resolve conflicts

Each data source node needs its own EA Export node, so if your data is scattered among a number of different sources, you might have a stream like this.

Figure 2-7
Exporting data into the repository from multiple data sources



You can choose to read the records from one, some, or all of your data sources if you have more than one. The Entity Analytics system analyzes the records you select and adds an identifier field named \$EA_ID to each one. Where two or more records that relate to previously ambiguous identities can now be resolved, the identifiers added to those records are unique throughout the repository. The system also adds a field to show the data source from which the record originates.

You connect each data source node to its own EA Export node, map the input fields to the repository features, and then run the stream to export the data from SPSS Modeler into the repository and resolve any identity conflicts, in a single operation. To illustrate how this works, suppose that you have the following records in four different data sources.

External data

Name	Phone	Credit Risk
Mike	555-1234	560
Joe	555-4567	780

Employees

Name	Address	Phone
Michael	1234 5th Street	555-1234
Fred	543 1st Avenue	555-9876

Customers

Name	Address	Savings
Susan	1234 5th Street	\$1234
Joe	777 Oak Street	\$5

Loans

Name	Address	Phone	Loan
Sue	1234 5th Street	555-1234	\$10,000
Joseph	777 Oak Street	555-4567	\$50,000

As we have seen, you export each data source into the repository in turn. As you do so, the repository updates the resolution of each record. In the repository, each record is preceded with an identifier field (named *SEA-ID*) and a source indicator field (named *SEA-SRC*), which shows the data source where the record originated. So in our example, when you have exported all four data sources, the repository contents look like this.

Table 2-1

Example of repository contents after export stage

SEA-ID	SEA-SRC	Name	Phone	Address	Credit Risk	Savings	Loan
1	Employees	Michael	555-1234	1234 5th St			
1	External	Mike	555-1234		560		
2	Customers	Joe		777 Oak St		\$5	
2	External	Joe	555-4567		780		
2	Loans	Joseph	555-4567	777 Oak St			\$50,000
3	Employees	Fred	555-9876	543 1st Ave			
4	Customers	Susan		1234 5th St		\$1234	
4	Loans	Sue	555-1234	1234 5th St			\$10,000

The Entity Analytics system has determined that *Mike* in the *External* data set is the same person as *Michael* in the *Employees* data set based on a telephone number in common, and assigned him the ID 1.

The case of *Joe* in the *External* data set is a little more difficult. Is he the same person as the *Joe* in *Customers*? It's impossible to tell from just those two data sources, but we have a third source, *Loans*, containing a *Joseph*. Now we have a match: *Joseph's* telephone number is the same as that of *Joe* in the *External* data set. On this basis, the system determines that they are the same person and gives them the identifier 2.

There are no multiple records for *Fred*, so he is given ID 3. *Susan* from *Customers* is identified as the same person as *Sue* from *Loans* as they have the same address, so she is assigned ID 4.

Note: This is an example of optimistic matching for the purposes of illustration. You could choose a more pessimistic rule set, so that a simple name and telephone number or address by itself would not constitute an exact match, and assign both records the same identifier.

Stage 6: Analyzing the resolved identities

Having resolved the identity conflicts in the repository, you can now perform further analysis and processing on the results. For example, if you suspect possible fraudulent activity with the existence of duplicate records for the same identity, you might want to produce a report listing the duplicates.

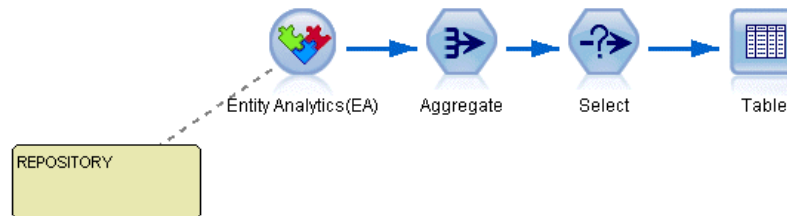
You start by creating an Entity Analytics(EA) source node and linking it to the repository.

The output from the node consists of the following fields.

- The identifier field added by the system (*\$EA-ID* in the Stage 5 example)
- The source indicator field added by the system (*\$EA-SRC* in the Stage 5 example)
- The unique key field that you designated in Stage 4

To view the output in SPSS Modeler, you can attach an SPSS Modeler output node such as a Table node or a Report node and run this part of the stream. If you need to summarize the output, which could be very large, you can include record operations nodes such as Aggregate or Select nodes, as in the following illustration.

Figure 2-8
Viewing the output



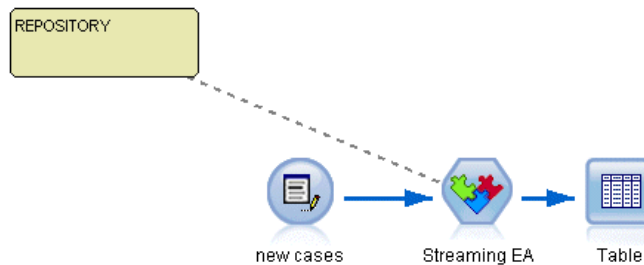
The Entity Analytics(EA) source node is fully described later. For more information, see the topic [Analyzing the resolved identities \(Entity Analytics\(EA\) source node\)](#) in Chapter 3 on p. 32.

Stage 7: Resolve new cases against the repository

You have resolved the identities of all the records in all of your data sources. But what happens when you want to compare a set of new records to see how they relate to what you know, for better scoring? This is where the Streaming EA node comes in.

First, you add a new SPSS Modeler data source node to read your new data into the stream. Next, you connect this source node to a Streaming EA node. To view the output, you add a Table node as before, so that this part of the stream now looks like this.

Figure 2-9
Resolving new cases



When you run this part of the stream, the Streaming EA node reads each new record and compares it against the repository contents. If it finds matching records in the repository, the Streaming EA node outputs all matching records together with the new record, to which it adds the ID and source indicator fields. If no match is found, the process node outputs only the new record with the ID and source indicator fields added.

To illustrate this, suppose that the repository currently consists of the contents that were output by the Entity Analytics(EA) source node. See [Table 2-1](#) on p. 10.

Now we receive the following new records. Do they relate to anyone we already know?

Table 2-2
New records to be scored

Name	Address	Phone	Loan
Suzan	1234 5th Street	555-1234	\$100,000
Mark	888 9th Ave	555-9999	\$60,000

Comparing the new data with the existing repository contents, the Streaming EA node matches the first new record with the person having the identifier 4 in the existing records. However, no match can be found for the second new record, so this is assigned a new, unique identifier, 5.

The Streaming EA node adds the identifier and source indicator fields, and outputs the new records together with all their matching records. Thus the output will be as follows.

Table 2-3
Output from the Streaming EA node

\$EA-ID	\$EA-SRC	Name	Phone	Address	Credit Risk	Savings	Loan
4	Cust	Susan		1234 5th St		\$1234	
4	Loan	Sue	555-1234	1234 5th St			\$10,000
4	New Loan	Suzan	555-1234	1234 5th Street			\$100,000
5	New Loan	Mark	555-9999	888 9th Ave			\$60,000

This output can then be aggregated by entity analytics identifier and passed to other downstream nodes for further processing.

The Streaming EA node is fully described later.

Stage 8: Generate alerts

Once again, potentially suspicious activity may become apparent. In this case, the person with identifier 4 already has a loan of \$10,000 and is now applying, under a slightly different name, for another one ten times that size. This may, of course, be perfectly acceptable and not done with any fraudulent intent. However, if such activity counts as suspicious according to your business rules, it may be worth looking into.

You could, for example, attach and run a SPSS Modeler Table node or a Report node, print off the contents of its output window, and have someone read it and generate alerts manually. Alternatively, you could pass the output of the Streaming EA node to a risk assessment model that you had previously created in IBM® SPSS® Modeler, producing a set of scores that more closely reflects your business rules. Another possibility is to export the output to a database or some other medium for further processing. With SPSS Modeler, you have a wide choice of actions to suit your particular requirements.

Entity analytics tasks

About the tasks

This section describes the following entity analytics tasks.

- Setting up an entity repository
- Configuring an entity repository
- Analyzing the resolved identities
- Resolving new cases against the entity repository
- Purging an entity repository
- Deleting an entity repository
- Using entity analytics with other IBM SPSS products
- Administering entity analytics

Setting up an entity repository (EA Export node)

The process of setting up an entity repository consists of the following tasks.

1. Connect to a data source. For more information, see the topic [Connecting to a data source](#) on p. 15.
2. Create the repository. For more information, see the topic [Creating the repository](#) on p. 16.
3. Map input fields in the data source to features in the repository. For more information, see the topic [Mapping input fields to features \(EA Export node\)](#) on p. 19.

When you have set up the mappings, you can display them either for the current data source, or for all data sources that are known to the repository. For more information, see the topic [Displaying the field mappings \(EA Export node\)](#) on p. 21.

The entity repository

The repository provides a central storage area, acting as a data cache for all of the entity information. Because the repository is live, it has a single state, so there is no concept of versioning with an entity repository. The repository holds the current state of all the input data, so it can grow very large.

You can maintain the repository contents by means of an easy-to-use graphical interface. For more information, see the topic [Configuring an entity repository](#) on p. 22.

Note: The version of IBM® SPSS® Modeler Entity Analytics supplied with IBM® SPSS® Modeler Premium supports a single repository hosted on the IBM solidDB product that is bundled with IBM SPSS Modeler Entity Analytics. With this version, you must delete an existing

repository before creating a new one. A separately-licensed upgrade to IBM SPSS Modeler Entity Analytics (known as IBM SPSS Modeler Entity Analytics Unleashed) is available that allows more than one repository to co-exist on the same system; each repository can contain more than 10 million rows and use more than four processor cores. Contact your local IBM Support representative for details.

Connecting to a data source

You start by reading your source data into SPSS Modeler by means of a source node.

To connect to a data source

1. From the Sources tab on the nodes palette at the bottom of the SPSS Modeler main window, double-click an icon corresponding to the type of source data. Doing so adds a source node to the screen canvas.
2. On the screen canvas, double-click the icon to open its dialog box.
3. In the File field, enter the location and name of the source data file.
4. Complete the rest of the dialog box as necessary (click the Help button for more information), then click OK.
5. If the source data file does not have a field that uniquely identifies each record, add one by means of a Derive node. For more information, see the topic [Adding a unique record identifier](#) on p. 15.

Adding a unique record identifier

Each data source file that is input to the entity repository must have one field that uniquely identifies each record. If a data source file does not have such a field, you can add one by means of an SPSS Modeler Derive node.

To add a unique record identifier to a data source file

1. On the screen canvas, click the source node that you added in the previous task.
2. From the Field Ops tab on the nodes palette, double-click the Derive icon to attach a Derive node to the source node.
3. On the screen canvas, double-click the Derive node to open its dialog box.
4. In the Derive field, replace the default name with a meaningful name (such as ID) for the identifier field you are adding.
5. Ensure that the Derive as field is set to Formula
6. Set Field type to Continuous.
7. In the Formula text box, type @INDEX and click OK.

Creating the repository

You need to create a repository to store all the input data.

To create a repository

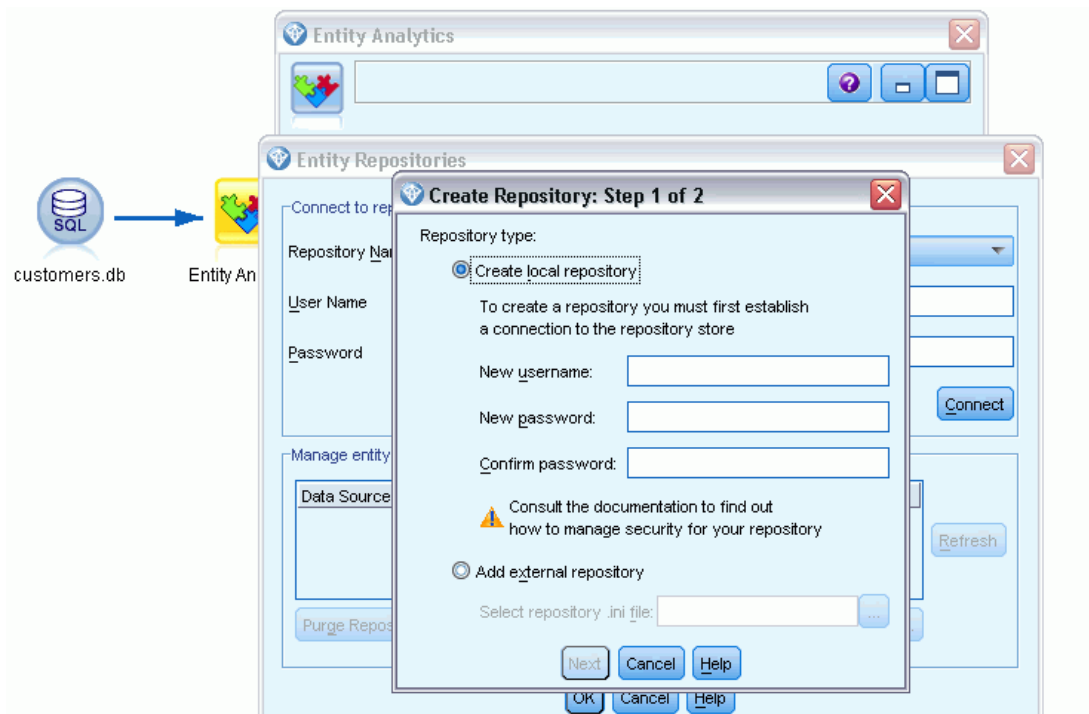
1. From the Export tab of the SPSS Modeler node palette, place an EA Export node onto the stream canvas.

Note: If you are creating a repository for the first time, use an EA Export node and connect it to the SPSS Modeler source node containing the data you want to input to the repository (or to the Derive node, if you added one to obtain a unique identifier field). To connect the nodes, do the following.

- Right-click the SPSS Modeler source node.
 - Choose Connect.
 - Click the EA Export node.
2. Double-click the EA Export node to open its dialog box.
 3. Click the Entity repository list.
 4. Click <Browse...> to display the Entity Repositories dialog box.
 5. On the Entity Repositories dialog box, click the Repository Name field.
 6. Choose <Create new repository...> to display the Create Repository wizard.

Create Repository wizard

Figure 3-1
Create Repository wizard



Step 1

Here you choose whether to create a local repository, using the IBM solidDB product that is bundled with IBM SPSS Modeler Entity Analytics, or to use an external database for the repository.

Create local repository. Specify the administrator username and password details for the IBM solidDB database that is to host the repository you are creating. Confirm the password and click Next.

If you subsequently need to change the administrator credentials, you do so by means of the command line editor for the database. For more information, see the topic [Managing administrator credentials for the repository database](#) on p. 41.

Note: Only one username and password combination is possible. All users logging in to the repository share the same username and password.

Add external repository. Use this option if you want to use an external database to host the repository. Type the location of the database .ini file in the Select repository .ini file field and click Next.

Step 2

New repository name. Type a unique name for the new repository.

Import configuration from. (Local repository only) If you want to base the configuration on that of an existing repository, choose the repository here, otherwise choose Default. For more information, see the topic [Configuring an entity repository](#) on p. 22.

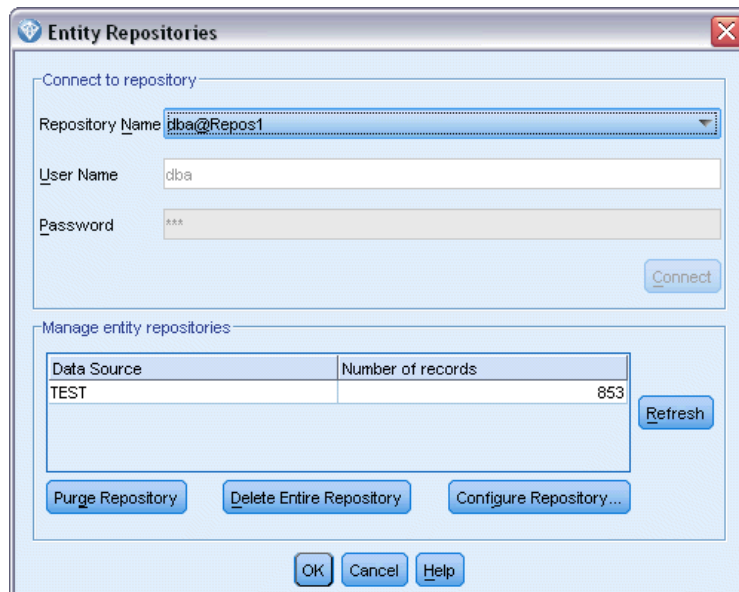
If you choose an existing repository, enter the connection details if they are different from the ones you entered on the previous screen.

Click OK to create the new repository and display the Entity Resolution Instances dialog box, from where you can connect to the repository.

Entity repository options

The Entity Repositories dialog box contains a number of options for creating, connecting to, configuring, and maintaining an entity repository.

Figure 3-2
Entity repository options



Connect to repository. Use these options to create a new entity repository, or to connect to an existing one.

- **Repository Name.** Shows the current entity repository if one exists. To choose a different repository if more than one exists, select one from the list.

To create a new repository, select <Create New Repository...>. Doing so starts a wizard that guides you through the creation process.

- **User Name.** Enter your username for the selected repository.

- **Password.** The password for this username.
- **Connect.** Click to connect to the current repository.

Manage entity repositories. The table lists the data sources that have been loaded into the current repository (the one to which you are connected), showing the number of records in each data source.

- **Refresh.** Updates the data source and size information in the table, for example if you have added a new data source or changed the size of an existing one.
- **Purge Repository.** Removes all the source data from the repository but maintains all the configuration details. You can use this option if the configuration information is still useful, but you want to remove all the data records from the repository. For more information, see the topic [Purging an entity repository](#) on p. 43.

Delete Entire Repository. Completely destroys the current repository contents and configuration details. For more information, see the topic [Deleting an entity repository](#) on p. 44.

Configure Repository. Displays a window where you can configure the current repository. For more information, see the topic [Configuring an entity repository](#) on p. 22.

Mapping input fields to features (EA Export node)

The repository provides a number of predefined features as standard. Different data sources may use different field names (for example, Address1 or Address Line 1) for information types that correspond to the same feature. To avoid duplication, it is necessary to map input data source fields to specific repository features. You don't need to map every field in the data set, just the ones that are likely to correspond to the same feature in other data sets.

Where a data source uses fields corresponding to other types of information that are not predefined in the repository, you can create new features from the Repository Configuration window. For more information, see the topic [Configuring an entity repository](#) on p. 22.

To map input fields to features

1. Attach an EA Export node to a data source node on the stream canvas. Each data source node that you use must be attached to its own EA Export node.
2. Open the EA Export node to display the Inputs tab, which contains options for mapping the input fields. For more information, see the topic [Repository input options for mapping](#) on p. 20.
3. On the EA Export node, select the Repository tab to view the mapping assignments for either the current data source, or for all data sources if you are using more than one.
4. To save a set of mapping assignments (for example, for use with another data source node), click Export Mapping.

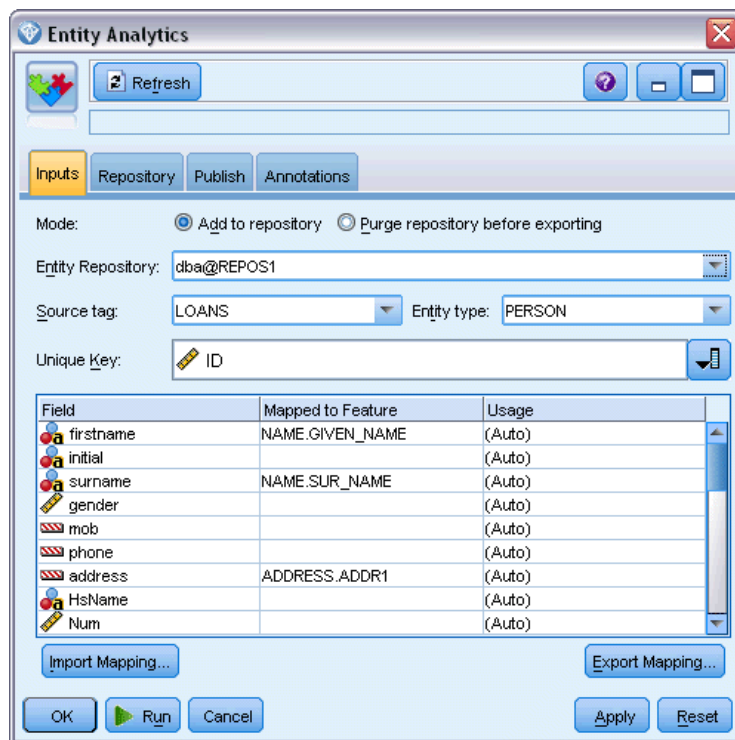
When you have finished mapping the first data source node, repeat the process for any other data source nodes that you want to use.

Repository input options for mapping

The Inputs tab contains the options for mapping data source fields to repository features ready for export to the repository. Set up the mapping assignments on this tab, optionally click the Repository tab to see the mapping for other data sources, then click Run to export the data to the repository.

If you have already stored a set of mappings in an XML file, you can use them by clicking Import Mapping.

Figure 3-3
Mapping input fields to features



Mode. Leave the default selection, Add to repository, if you want to add the source file records to the existing contents of the repository. If you want to clear out the repository contents but maintain the configuration information before adding the source records, choose Purge repository before exporting.

Entity repository. Shows the current entity repository if one exists. To choose a different repository if more than one exists, select one from the list. To create a new repository, choose <Browse...> to display a dialog box from where you can create the repository. For more information, see the topic [Entity repository options](#) on p. 18.

Source tag. A list of tags indicating data sources currently known to the repository. Choose one from the list, or choose <Add new source tag...> to create a tag for a new data source.

Entity type. A list of entity types (that is, sets of features) defined in the repository. Choose one from the list, or choose <Add new entity type...> to display the repository configuration window, where you can define a new entity type. For more information, see the topic [Configuring an entity repository](#) on p. 22.

Unique key. (required) The input field to use for the unique identifiers for the data records.

Mapping table. In this table you can map each input field to a corresponding feature in the repository. If a suitable feature does not exist in the selected entity type, you can create a new feature here.

- **Field.** The set of input fields in the selected data source. Each field has an icon indicating the measurement level (that is, the data type) for the field.
- **Mapped to Feature.** To map a field to a feature, double-click this column (or press the space bar) on the field row and choose a feature from the list. If a suitable feature is not available, choose <Add new feature...> to display the repository configuration window, where you can define a new feature for this entity type. For more information, see the topic [Configuring an entity repository](#) on p. 22.
- **Usage.** Indicates the context of a particular field where more than one context is possible, for example, home and work telephone numbers. There are preset usage types available for the ADDRESS and PHONE features, and you can create your own usage types for all features. To set a usage other than the default (Auto), click this column on the row you want, and either choose one of the existing usage types (if any), or click <Add usage...> to create a new one. For more information, see the topic [Maintaining the entity types](#) on p. 28.

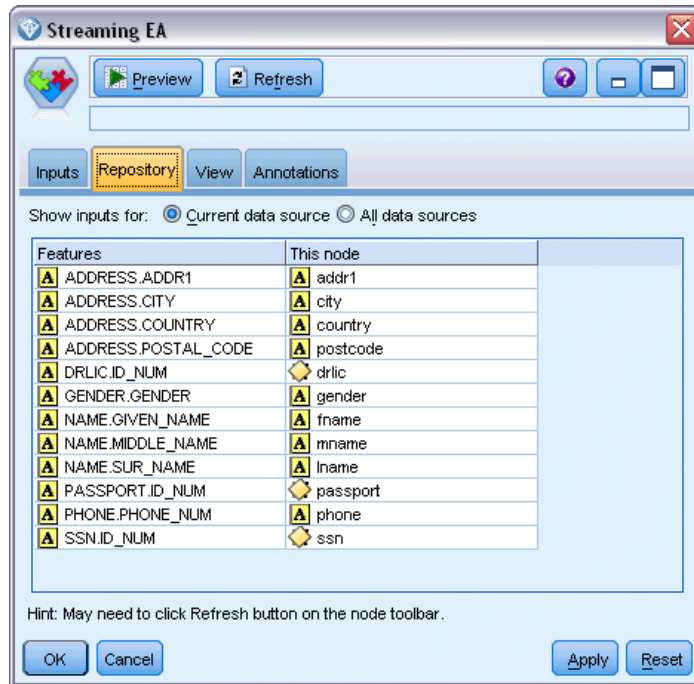
Import mapping. Imports a previously exported set of field-to-feature mappings from an external XML file. This can be useful if you have different data sources with the same mapping requirements, as it avoids having to redefine the same mappings for the different sources.

Export mapping. Exports to an external XML file the set of field-to-feature mappings shown in the mapping table.

Displaying the field mappings (EA Export node)

On the Repository tab, click the Refresh button to see which repository features have input fields mapped to them. You can see this either for the current data source (the one controlled by the source node attached to this export node), or for all data sources.

Figure 3-4
Displaying the field mappings



Show inputs for. Choose an option to display the mappings for either the current data source, or for all data sources that are known to the repository.

Refresh. Updates the display for the selected input option.

Features. A list of all the features that have mappings in the displayed data sources. Unmapped features are not shown.

<Data source>. Each column lists the mapped fields in a particular data source for each feature for which a mapping has been defined.

Configuring an entity repository

You can maintain the repository contents from the Repository Configuration window, which provides an easy-to-use visual interface to the entire repository.

If you will be using more than one repository with the same or similar configurations, you can set up a basic configuration and export it to a file that you can then import into other repositories. For more information, see the topic [Reusing a repository configuration](#) on p. 31.

Caution: If you modify and save the configuration of a repository that already contains data, you are strongly advised to purge the repository contents and reload the data. Doing so avoids the repository being left in an inconsistent state.

To set up a repository configuration

1. Open any Entity Analytics node.
2. Click the Entity repository list.
3. Click <Browse...> to display the Entity Resolution Instances dialog box.
4. On the Entity Resolution Instances dialog box, click the Repository Name list.
5. Select the repository for which you want to set up the configuration.
6. If you are not already connected, enter the administrator username and password and click Connect.
7. When the Configure Repository button is enabled, click it to display the Repository Configuration window.
8. Create the configuration details as explained in the following sections.

The navigation pane on the left side of the Repository Configuration window contains a tree structure from where you can manage the different characteristics of the repository.

Table 3-1

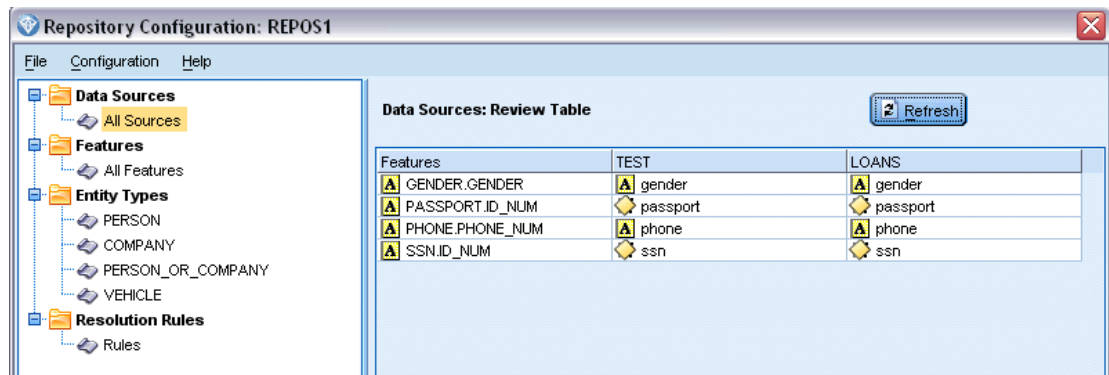
Main elements of the Repository Configuration window

Section	Description	
Data Sources	Display the mappings of all the data sources to the various repository features.	For more information, see the topic Viewing the data source mappings on p. 23.
Features	Create a new feature, or duplicate, edit or delete an existing one.	For more information, see the topic Maintaining the repository features on p. 24.
Entity Types	Create a new entity type, or manage existing ones (duplicate, rename, attach or remove features, delete).	For more information, see the topic Maintaining the entity types on p. 28.
Resolution Rules	Set threshold for entity matching.	For more information, see the topic Setting the threshold for entity matching on p. 30.

Viewing the data source mappings

In the Data Sources section of the Repository Configuration window, the All Sources entry provides a read-only display of the mappings of all the data sources to the various repository features.

Figure 3-5
Data source mappings in the configuration window



Click Refresh to update the list if new data sources have been added to the repository.

Note: You cannot add a data source to the repository here. Data sources can be added only by creating an SPSS Modeler source node and connecting it to an Entity Analytics Export node. For more information, see the topic [Connecting to a data source](#) on p. 15.

Maintaining the repository features

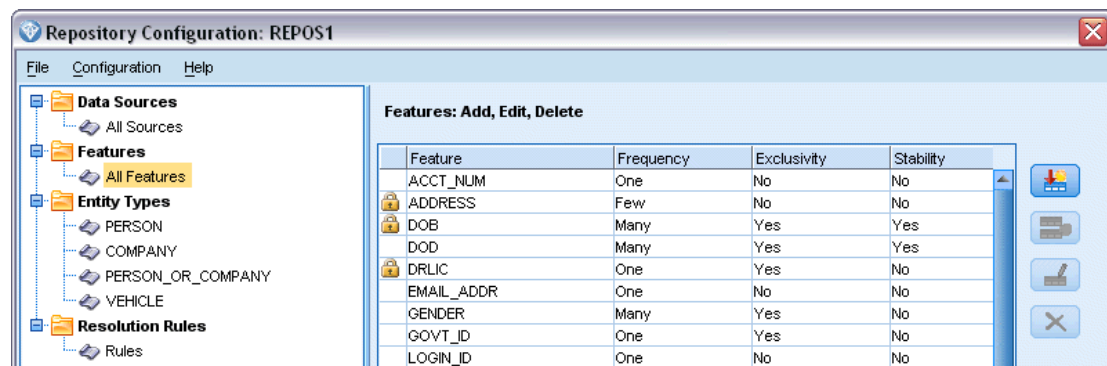
A repository feature is an individual information type that can be used with an entity data source. Some features (for example, First Name, Last Name, Date of Birth, and so on) may be capable of being used with many different data sources, while other features will be specific to a particular data source. A feature can contain one or more elements; each element is typically the equivalent of a field in a data record, or a column in a database table.

In the Features section of the Repository Configuration window, the All Features entry provides the means to maintain all the repository features. You can do the following.

- Create a new feature
- Duplicate an existing feature (for example, to create a new feature based on an existing one)
- Edit an existing feature
- Delete an existing feature

Instructions for these tasks are given later in this section.

Figure 3-6
Features listed in the configuration window



The feature list shows all the features that have been defined in this repository. The columns in the list show the various properties a feature may have.

Feature. The name of the feature. A padlock symbol next to a feature name indicates that the feature is locked. Features that are locked cannot be deleted or duplicated, nor can changes to them be saved.

Frequency. Indicates how many identities can have the same value for this feature. Valid values are None, Name, One (e.g. for a passport number), Few (e.g. for an address), or Many (e.g. for a date of birth).

Exclusivity. Indicates that an entity should typically have only one of this type of feature. For example, a date of birth or a national identity number would have the value Yes here, while an address or credit card number would have the value No (as an entity might have multiple addresses or credit cards).

Stability. Indicates the stability value of this feature (that is, whether it is *unlikely* to change during the lifetime of an entity). For example, a date-of-birth feature would have the value Yes as it never changes, but an address feature would have No as it is quite likely to change, and is therefore less stable. *Note:* Gender is typically stable throughout a lifetime, but because it is frequently incorrectly specified owing to bad data, the default configuration gives it the value No.

To create a new feature

- Do one of the following.
 - Click the Create New Feature button (the top button on the right of the screen).
 - Right-click All Features in the navigation pane on the left of the screen, and choose New Feature.
- Complete the Add/Edit Feature dialog box. For more information, see the topic [Adding or editing a feature](#) on p. 26.

To duplicate an existing feature

1. In the Feature column of the table on the right of the screen, select the feature you want to duplicate.
2. Click the Duplicate Selected Feature button (the second button on the right of the screen).
3. Complete the Add/Edit Feature dialog box. For more information, see the topic [Adding or editing a feature](#) on p. 26.

To edit an existing feature

Caution: If you edit or delete a feature or feature element when the repository already contains data, you should subsequently purge the repository and reload the data. Doing so avoids leaving the repository in an inconsistent state.

1. In the Feature column of the table on the right of the screen, select the feature you want to edit.
Note: You can edit only those features that you have created, not the system-supplied features.
2. Click the Edit Selected Feature button (the third button on the right of the screen).
3. Complete the Add/Edit Feature dialog box. For more information, see the topic [Adding or editing a feature](#) on p. 26.

To delete an existing feature

Caution: If you edit or delete a feature or feature element when the repository already contains data, you should subsequently purge the repository and reload the data. Doing so avoids leaving the repository in an inconsistent state.

1. In the Feature column of the table on the right of the screen, select the feature you want to delete.
Note: You can delete only those features that you have created, not the system-supplied features.
2. Do one of the following.
 - Click the Delete Selected Feature button (the bottom button on the right of the screen).
 - Right-click All Features in the navigation pane on the left of the screen, and choose Delete.
3. Click Continue to confirm deletion of the feature. *Note:* You cannot undo deletion of a feature.

Adding or editing a feature

Caution: If you edit or delete a feature or feature element when the repository already contains data, you should subsequently purge the repository and reload the data. Doing so avoids leaving the repository in an inconsistent state.

On the Add/Edit Feature dialog box, you can create a new repository feature, or duplicate or edit an existing feature.

Figure 3-7
Editing a feature

Element	Description	Data Type	Search	Compare	Descriptive
1 ADDR_TYPE	Address t...	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2 FULL_ADDR	Full address	String	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3 ADDR1	Addr line 1	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4 ADDR2	Addr line 2	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5 ADDR3	Addr line 3	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6 CITY	City	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
7 STATE	State	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
8 POSTAL_...	Postal code	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
9 COUNTRY	Country	String	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Feature type. A label indicating the type of information to which the feature relates. This label forms the first part of the feature identifier.

Description. A brief text description of the feature type, for information only.

Frequency. Indicates how many identities can have the same value for this feature. Valid values are None, Name, One (e.g. for a passport number), Few (e.g. for an address), or Many (e.g. for a date of birth).

Exclusivity. Indicates that an entity should typically have only one of this type of feature. For example, a date of birth or a national identity number would have the value Yes here, while an address or credit card number would have the value No (as an entity might have multiple addresses or credit cards).

Stability. Indicates the stability value of this feature (that is, whether it is *unlikely* to change during the lifetime of an entity). For example, a date-of-birth feature would have the value Yes as it never changes, but an address feature would have No as it is quite likely to change, and is therefore less stable. *Note:* Gender is typically stable throughout a lifetime, but because it is frequently incorrectly specified owing to bad data, the default configuration gives it the value No.

Persist history. If set to Yes (default), preserves the history of changes to the field value. This is useful, for example, in the case of the address field for a customer. In this case, preserving the change history could improve matching if a customer has changed addresses a few times. However, for a field such as an account balance, you would set this to No to avoid preserving excessive amounts of data.

Elements table. A list of the elements that this feature comprises.

- **Element.** The element name.
- **Description.** A brief description of what the element provides.
- **Data Type.** The type of data that can be used for this element.
- **Search.** If selected, means that this element of the feature can be searched for by itself or in combination with other elements of the feature labeled search. For example, suppose that a PASSPORT feature has two elements, ID_NUM and COUNTRY. If you want to be able to search on ID_NUM alone, you select its Search box and clear the Search box for COUNTRY. Any modifiers to the feature, such as country or state, or issue and expiration date, should not be part of the search.
- **Compare.** Indicates which elements of the feature will be sent to the comparison routine for entity resolution. Continuing with the example of a PASSPORT feature with two elements, ID_NUM and COUNTRY, since COUNTRY may not always be supplied, only ID_NUM should be sent to the comparison routine.
- **Descriptive.** Indicates which elements of the feature should appear in its internal description. In the previous example of a PASSPORT feature with two elements, ID_NUM and COUNTRY, if you wanted both to appear in the description of the feature, you would select the Descriptive box for both. But suppose that the feature had an additional element, ISSUE_DT. You would clear its Descriptive box to ensure that it did not appear in the feature description.

Add New Element button. Adds a new row to the elements table, so that you can define a new element.

Delete Element button. Deletes a selected row from the elements table. You cannot undo this operation.

Caution: If you edit or delete a feature or feature element when the repository already contains data, you should subsequently purge the repository and reload the data. Doing so avoids leaving the repository in an inconsistent state.

Maintaining the entity types

An **entity type** is a named set of repository features that logically belong together. For example, an entity type intended for use with a customer data set might consist of features such as Name, Date of Birth, Gender, Address, Telephone Number, and so on.

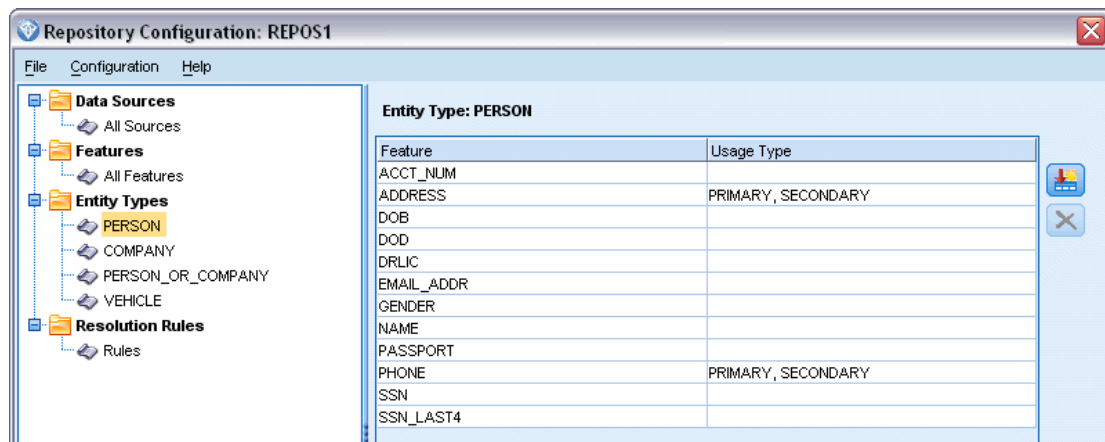
The IBM SPSS Modeler Entity Analytics repository is supplied with a standard set of entity types, and you can add your own.

The Entity Types section of the Repository Configuration window lists the different entity types that have been created. You can do the following.

- Create a new entity type
- Duplicate an existing entity type (for example, to create a new entity type based on an existing one)
- Attach features to an entity type
- Remove features from an entity type

- Rename an entity type
- Delete an entity type

Figure 3-8
Listing the features belonging to an entity type



Entity Type. The name of the selected entity type.

Feature. The list of valid features that this entity type comprises.

Usage Type. (Optional) Indicates different contexts in which this feature might be used. Double-click this column to add or edit a usage type, separating usage types with a comma and a space. The values you specify here define the values displayed on the EA Export or Streaming EA nodes when a user clicks the Usage column for a feature on the Inputs tab. For more information, see the topic [Repository input options for mapping](#) on p. 20.

To create a new entity type

1. Right-click Entity Types in the navigation pane on the left of the screen.
2. Choose New Entity Type.
3. Enter a unique name for the entity type and click OK.
4. Attach features to the entity type (see next section).

To attach features to an entity type

1. Select the entity type in the navigation pane on the left of the screen.
2. Click the Attach Feature button (the top button on the right of the screen).
3. From the list of available features, choose one or more (use Ctrl-click to choose multiple features) and click OK.

To remove features from an entity type

1. Select the entity type in the navigation pane on the left of the screen.

2. Select one or more features from the table of attached features on the right of the screen. Use Ctrl-click to choose multiple features.
3. Click the Detach Feature button (the bottom button on the right of the screen).

To duplicate an existing entity type

1. In the navigation pane on the left of the screen, right-click the entity type you want to duplicate.
2. Choose Duplicate Entity Type.
3. Enter a unique name for the new entity type and click OK.
4. Attach features to, or remove features from, the entity type as required (see earlier instructions).

To rename an entity type

Caution: If you edit or delete a feature or feature element when the repository already contains data, you should subsequently purge the repository and reload the data. Doing so avoids leaving the repository in an inconsistent state.

1. In the navigation pane on the left of the screen, right-click the entity type you want to rename.
2. Choose Rename.
3. Enter the new name for the entity type and click OK.

To delete an entity type

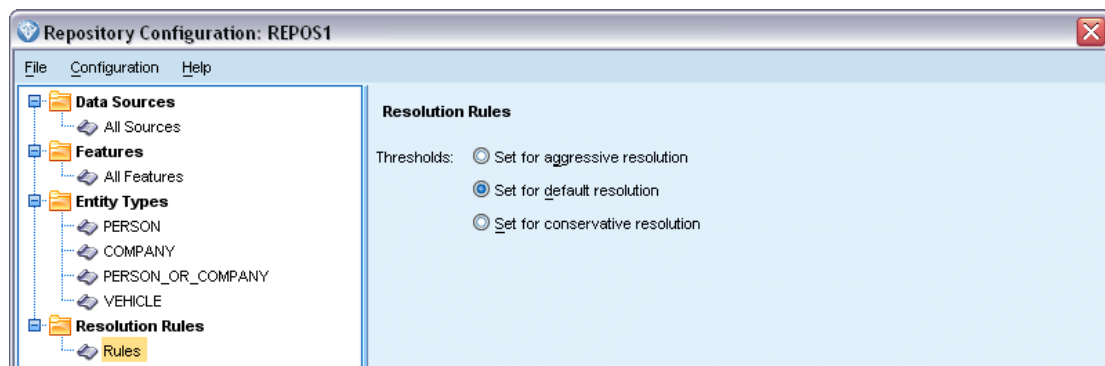
Caution: If you edit or delete a feature or feature element when the repository already contains data, you should subsequently purge the repository and reload the data. Doing so avoids leaving the repository in an inconsistent state.

1. In the navigation pane on the left of the screen, right-click the entity type you want to delete.
2. Choose Delete.
3. Click OK to confirm deletion of the entity type. *Caution:* You cannot undo deletion of an entity type.

Setting the threshold for entity matching

In the Resolution Rules section of the Repository Configuration window, you choose the threshold at which entity matching will occur.

Figure 3-9
Setting the entity resolution rules



When you create the repository, matching is preset to the default threshold.

Choose Set for aggressive resolution if you are not finding sufficient matches in your records to perform entity resolution.

Choose Set for default resolution to return to the default threshold from one of the other settings.

Choose Set for conservative resolution if you are finding too many matches.

Reusing a repository configuration

If you have already set up a configuration and you want to use it for another repository, you can export the existing configuration to an XML file and import the file into the other (target) repository.

To reuse an existing configuration

1. Display the Repository Configuration window for the repository whose configuration you want to use. For more information, see the topic [Configuring an entity repository](#) on p. 22.
2. From the menu in that window, choose Configuration > Export Configuration.
3. In the Save As dialog box, choose the name and location of the export XML file.
4. Display the Repository Configuration window for the target repository.
5. From the menu in that window, choose Configuration > Import Configuration.
6. In the Open dialog box, choose the name and location of the previously-exported XML file, and click Open .

Saving your configuration changes

To save the changes to the configuration

From the menu in the Repository Configuration window, choose File > Save.

Closing the configuration window

To exit from the configuration window

From the menu in the Repository Configuration window, choose File > Exit.

If you have unsaved changes to the configuration, click OK to save the changes and exit, or Cancel to exit without saving.

Analyzing the resolved identities (Entity Analytics(EA) source node)

With at least one data source input to the repository, you can use the Entity Analytics(EA) source node to pass the resolved identities to other IBM® SPSS® Modeler nodes for further analysis or processing, such as creating a report listing the resolved identities.

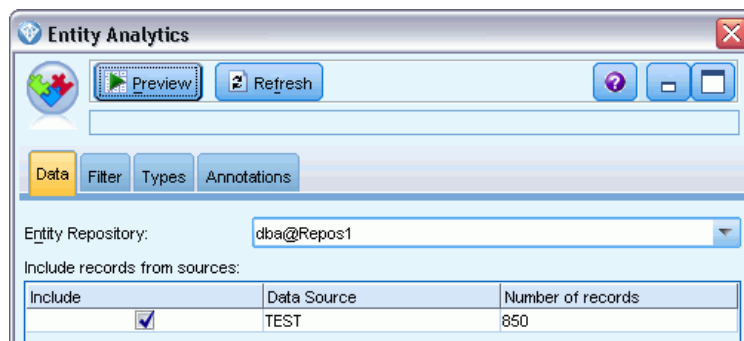
To analyze the resolved identities

1. Add an Entity Analytics(EA) source node to a stream.
2. Open the Entity Analytics(EA) node.
3. On the Data tab, select the entity repository and one or more of its input data sources (click Refresh to update the record counts). For more information, see the topic [Selecting a data source](#) on p. 32.
4. Add further nodes to the stream to perform the processing you want. For more information, see the topic [Adding nodes to the stream](#) on p. 34.

Selecting a data source

On the Data tab, you select at least one data source in the repository on which to perform further processing. Click Refresh to update the record counts for the data sources listed.

Figure 3-10
Selecting a data source on the Source node



Entity repository. Shows the current entity repository if one exists. To choose a different repository if more than one exists, select one from the list. To create a new repository, choose <Browse...> to display a dialog box from where you can create the repository. For more information, see the topic [Entity repository options](#) on p. 18.

Include records from sources. This table lists the different data sources that have been input to the repository, together with the number of records in each source. Select the Include check box for those data sources you want to use for performing further analysis and processing.

Renaming data fields

You can use the Filter tab to rename any of the resolved identity fields that are passed downstream for further processing. You might want to rename a resolved identity field, for example, to maintain field name compatibility when merging downstream with another data set.

The fields with their original names are as follows.

Table 3-2
Resolved identity fields

Field	Description
\$EA-ID	Entity identifier
\$EA-SRC	Source tag identifying the data source where the records originated
\$EA-KEY	Field designated as unique key in data source file

Note: Although you can also use the Filter tab to filter out fields, you should not do so here as the resolved identity fields are the bare minimum needed for entity analytics processing.

Setting type information for data fields

On the Types tab, you can view or change various properties of the resolved identity fields that are passed downstream for further processing.

The properties that you can change are the same as those on the Types tab of a regular SPSS Modeler Type node, and are as follows.

Table 3-3
Type properties for fields

Property	Description
Measurement	The measurement level (that is, the data type), used to describe characteristics of the data in the field.
Values	Provides options for reading data values from the dataset.
Missing	Used to specify how missing values for the field will be handled.
Check	Validation options for ensuring that field values conform to the specified values or ranges.
Role	Specifies how the field will be used if the data is passed to a modeling node or model nugget.

Adding nodes to the stream

You can add various SPSS Modeler nodes to the stream to perform analysis or processing operations on the output from the Entity Analytics(EA) source node. For example, you could add one or more of the following.

- Aggregate node to summarize the output, which might be very large
- Select node to select a subset of the output
- Table node to view the output from the Entity Analytics(EA) source node
- Report node to print the output in a report
- SPSS Modeler export node to export the output to a different format, such as a spreadsheet or database

For more information, see the sections on Record Operations, Output, and Export nodes in the *IBM SPSS Modeler Source, Process and Output Nodes Guide*.

Comparing new cases against the repository (Streaming EA node)

When you have already performed some identity resolution in the repository, you can use the Streaming EA node to compare new cases that you encounter subsequently with the repository contents. This node processes records from a new data source, compares them with the resolved entities already in the repository, and passes on any matching records for further processing. Matches can be set to be exact, or more loosely related to existing entities.

Like the EA Export node, the Streaming EA node takes a single SPSS Modeler source node as input. However, the Streaming EA node differs in the following way. Whereas the export node outputs records for all the entities related to its input records, the Streaming EA node outputs records for only those entities that relate to entities already resolved in the repository. For more information, see the topic [Output from the Streaming EA node](#) on p. 39.

To compare new cases against the repository

1. Connect to the data source containing the new records that you want to compare with the existing entities. For more information, see the topic [Connecting to a data source](#) on p. 15.

2. From the Record Ops tab, attach a Streaming EA node to the data source node.
3. Double-click the Entity Analytics Export node to open its dialog box.
4. Click the Entity repository list.
5. Click <Browse...> to display the Entity Repositories dialog box.
6. On the Entity Repositories dialog box, click the Repository Name field.
7. Click the name of the repository you want to use.
8. Enter the username and password for this repository and click Connect. Click OK when the repository is connected.
9. On the Streaming EA dialog box, select the Entity Type you want to use. For more information, see the topic [Maintaining the entity types](#) on p. 28.
10. Map the input fields in the data source to features in the repository. For more information, see the topic [Mapping input fields to features \(Streaming EA node\)](#) on p. 35.
11. Optionally click the Repository tab to see a table showing the mapping for this data source (or all the data sources that have been mapped). For more information, see the topic [Displaying the field mappings \(Streaming EA node\)](#) on p. 37.
12. Click the View tab to see details of the various data sources that have been input to the repository and set the selection criteria for retrieving existing entities. For more information, see the topic [Viewing the data sources \(Streaming EA node\)](#) on p. 38.
13. Click OK when the node is set up correctly.
14. Attach a Table node to the Streaming EA node and run the stream.

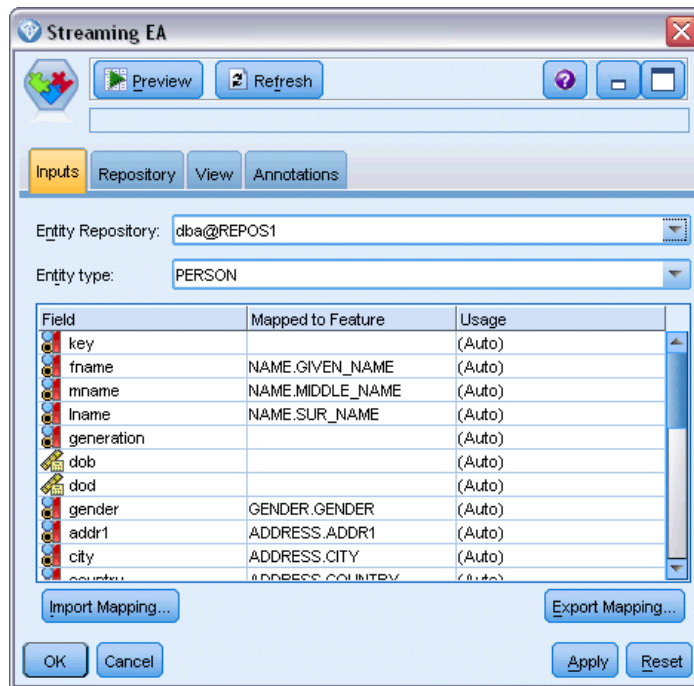
The output window of the Table node lists all the retrieved entities that match the new records in the data source. Output fields have the prefix \$EA- added. For more information, see the topic [Output from the Streaming EA node](#) on p. 39.

Mapping input fields to features (Streaming EA node)

The Inputs tab contains the options for mapping fields in the input to this node to repository features. Set up the mapping assignments on this tab, or select the View tab to see details of all the data sources in the repository, then click OK.

If you have already stored a set of mappings in an XML file, you can use them by clicking Import Mapping.

Figure 3-11
Mapping fields from new records to features in the repository



Entity repository. Shows the current entity repository if one exists. To choose a different repository if more than one exists, select one from the list. To create a new repository, choose <Browse...> to display a dialog box from where you can create the repository. For more information, see the topic [Entity repository options](#) on p. 18.

Entity type. A list of entity types (that is, sets of features) defined in the repository. Choose one from the list, or choose <Add new entity type...> to display the repository configuration window, where you can define a new entity type. For more information, see the topic [Configuring an entity repository](#) on p. 22.

Mapping table. In this table you can map each input field to a corresponding feature in the repository. If a suitable feature does not exist in the selected entity type, you can create a new feature here.

- **Field.** The set of input fields in the selected data source. Each field has an icon indicating the measurement level (that is, the data type) for the field.
- **Mapped to Feature.** To map a field to a feature, double-click this column (or press the space bar) on the field row and choose a feature from the list. If a suitable feature is not available, choose <Add new feature...> to display the repository configuration window, where you can define a new feature for this entity type. For more information, see the topic [Configuring an entity repository](#) on p. 22.
- **Usage.** Indicates the context of a particular field where more than one context is possible, for example, home and work telephone numbers. For more information, see the topic [Maintaining the entity types](#) on p. 28.

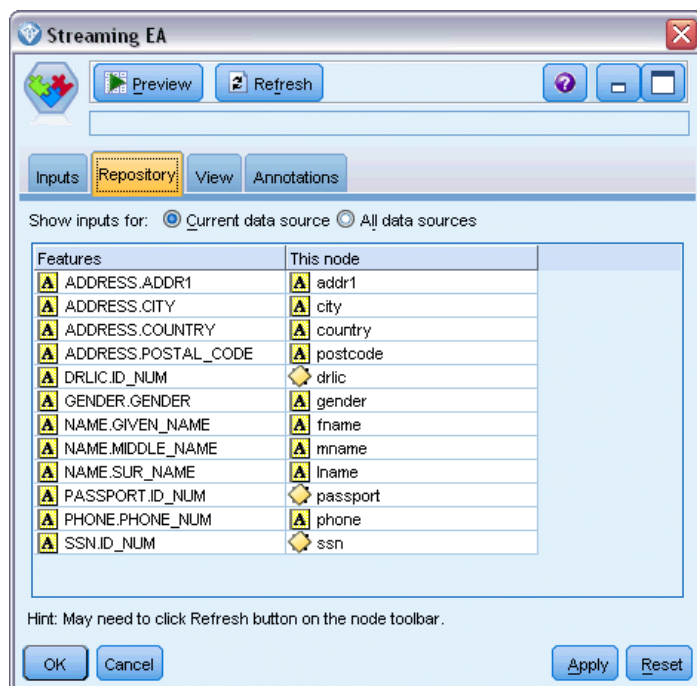
Import mapping. Imports a previously exported set of field-to-feature mappings from an external XML file. This can be useful if you have different data sources with the same mapping requirements, as it avoids having to redefine the same mappings for the different sources.

Export mapping. Exports to an external XML file the set of field-to-feature mappings shown in the mapping table.

Displaying the field mappings (Streaming EA node)

On the Repository tab, click Refresh to see which repository features have input fields mapped to them. You can see this either for the current data source (the one controlled by the source node attached to this export node), or for all data sources.

Figure 3-12
Displaying the field mappings



Show inputs for. Choose an option to display the mappings for either the current data source, or for all data sources that are known to the repository.

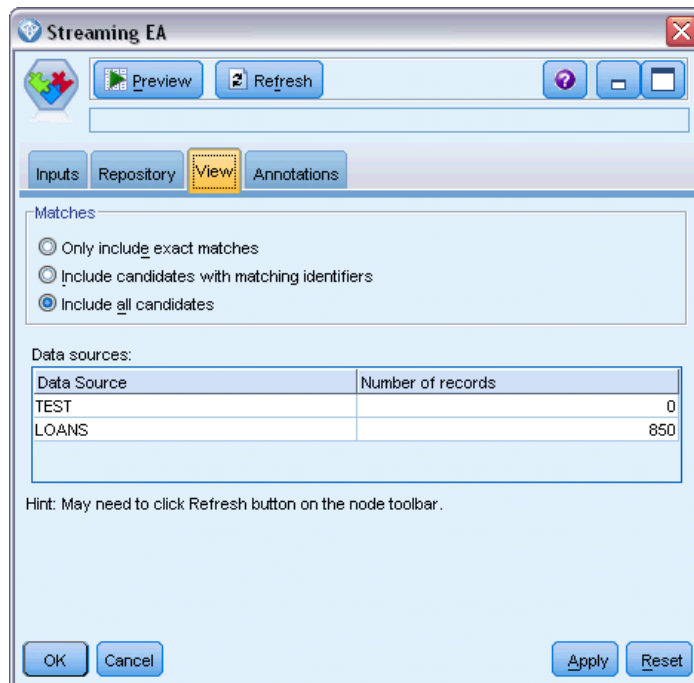
Features. A list of all the features that have mappings in the displayed data sources. Unmapped features are not shown.

This node. Each column lists the mapped fields in a particular data source for each feature for which a mapping has been defined.

Viewing the data sources (Streaming EA node)

On the View tab you can see details of the various data sources that have been input to the repository. These are the data sources against which the input to this node is processed, to search for and retrieve matching entities. Click Refresh to update the record counts.

Figure 3-13
Viewing the data source details



Matches. These options specify how closely the field-to-feature mapping information you specify on the Inputs tab is to be matched against the candidate records (that is, the entire repository contents). The closer the matching criterion, the fewer the entities that will be retrieved.

Note: If more than 20 matches are found, only the first 20 found will be returned.

- **Only include exact matches.** This is the closest matching criterion, and results in the fewest records being selected. Use this option when you want to return only those entities that are considered to be exact matches.
- **Include candidates with matching identifiers.** Use this setting when you want to return both matching entities and entities that share the same identifiers (those with features that have been configured with a frequency value of One, for example, matching credit card numbers, tax ID numbers, and so on).
- **Include all candidates.** Use this option when you want to see the widest possible number of entities in the repository that have shared features. This is the loosest matching criterion, and results in the greatest number of records being selected. This option returns exact matches and entities sharing almost any feature (typically those with a frequency value of One or Few). For example, both entities with the same tax ID number and entities with similar addresses would be included.

Data sources. This table lists the source tags of the different data sources that have been loaded into the repository, together with the number of records in each source.

Output from the Streaming EA node

The output from the Streaming EA node consists of the following fields for each record that is retrieved.

Field	Description
<i>Field1</i> [, <i>Field2</i> [, ... <i>FieldN</i>]]	Fields from the data source that contains the new records.
\$EA-ID	Entity identifier for this record in the repository.
\$EA-SC	Field indicating closeness of the match between this record and an observed entity in the repository; a value from 1.0 (poor match) to 10.0 (good match).
\$EA-SRC	Source tag identifying the data source where this record originated.
\$EA-KEY	Value of unique key for this record in data source file.
<i>\$EA-Feature1</i> [, <i>\$EA-Feature2</i> [, ... <i>\$EA-FeatureN</i>]]	Values of the mapped features for this record in the repository.

Using IBM SPSS Modeler Entity Analytics with other IBM SPSS products

Installers are available to enable you to use IBM SPSS Modeler Entity Analytics with the following products:

- IBM® SPSS® Collaboration and Deployment Services
- IBM SPSS Modeler Batch for Windows
- IBM SPSS Modeler Solution Publisher

You will need to run these installers before you can use the features of IBM SPSS Modeler Entity Analytics with these products. For more information, see the *IBM SPSS Modeler Premium Installation* guide.

Administrative tasks

For those repositories which are created within Entity Analytics, a new database service is created using the IBM solidDB product. There are a few administrative tasks associated with solidDB; these tasks are typically performed by the database administrator or the system administrator, and are:

- Configuring port assignments
- Managing administrator credentials for the repository database

Other administrative tasks that may need to be performed apply to all repositories, and are as follows.

- Moving the repository to a different storage directory
- Setting stream properties for date/time and timestamp fields
- Adjusting the timeout settings
- Running IBM SPSS Modeler Entity Analytics with SPSS Modeler client and SPSS Modeler Server on the same Windows system
- Purging an entity repository
- Deleting an entity repository
- Deleting a repository when unable to connect to it

Configuring port assignments

Each solidDB database service must be allocated a port which cannot be allocated to other services running on the machine. Database services reside on the same machine that runs IBM® SPSS® Modeler Server (or, when IBM® SPSS® Modeler is used without a connection to SPSS Modeler Server, the machine running IBM® SPSS® Modeler).

By default Entity Analytics assigns ports in the range 1320 to 1520, starting at port 1320 for the first repository created. In the event of a conflict, you can configure the assignment of ports by editing the file: `<modeler server installation path>/ext/bin/pasw.entityanalytics/ea.cfg` and setting appropriate values for the `min_port` and `max_port` settings. The default contents of this file are shown below:

```
# port range configuration for entity analytics

#

# this port range controls which ports SolidDB databases

# (created to store Entity Analytics Repositories in)

# may use. Configure this if the default port range will

# introduce a conflict on your system.

#

# default min_port = 1320

# default max_port = 1520

min_port, 1320

max_port, 1520
```

Managing administrator credentials for the repository database

The administrator username and password for the solidDB database that hosts an entity repository are defined when the repository is created. If you know the current credentials, you can change these details by means of the solidDB SQL editor.

To start the solidDB SQL editor

1. At a client machine, open a command prompt window.
2. Enter:

```
cd modeler_install_dir\ext\bin\pasw.entityanalytics\solidDB\bin
```

where *modeler_install_dir* is the directory where SPSS Modeler is installed.

3. Enter:

```
solsql -c "C:\Documents and Settings\All Users\Application  
Data\IBM\SPSS\Modeler\version\EA\repositories\repos_name
```

where *version* is the version number of the SPSS Modeler installation, and *repos_name* is the name of the repository.

4. At the prompts, enter the current database administrator username and password to display the solsql> prompt.

To change the database administrator password

1. At the solsql> prompt, enter:

```
alter user username identified by password;  
commit work;
```

where *username* is the current username of the database administrator and *password* is the new password.

2. Enter exit; to close the editor.
3. Restart the SPSS Modeler client.

To perform other administrative tasks

For information on other administrative tasks to do with the solidDB database, see the documentation for the appropriate version of IBM solidDB at <http://publib.boulder.ibm.com/>.

Moving the repository to a different storage directory

By default, the repository files are stored in a directory named *EA* at the following locations:

- C:\Documents and Settings\All Users\ApplicationData\IBM\SPSS\Modeler\version\EA (Windows systems)
- *modeler_install_directory*/ext/bin/pasw.entityanalytics/EA (UNIX systems)

As the files used to store the repository can become very large, you may need to move them to a different disk or partition to make more space available.

To move the repository to a different directory

1. Exit from SPSS Modeler.
2. Move the *EA* directory from the original location (as listed earlier) to a new location. For example, on Windows you might want to move it to a new location such as *F:\data\EA*.
3. Edit the file *<modeler server installation path>/ext/bin/pasw.entityanalytics/ea.cfg* to add the following option:

```
repository_data_directory, new_location
```

where *new_location* is the directory to which you moved the *EA* directory, for example *F:\data\EA*.

Setting stream properties for date/time and timestamp fields

If your source data includes fields containing date/time or timestamp data, ensure that the corresponding stream properties are set to the format recognized by IBM SPSS Modeler Entity Analytics.

To set the stream property format

1. On the main SPSS Modeler menu, choose:
Tools > Stream Properties > Options.
2. Select Date/Time.
3. Set Date format to YYYY-MM-DD.
4. Set Time format to HH:MM:SS.
5. Click OK.

Adjusting the timeout settings

On systems that are slow or heavily loaded, if you experience errors when creating or accessing repositories you may need to increase the timeout settings for starting and stopping the entity analytics engine or the entity analytics database server.

To adjust the timeout for the entity analytics engine

1. Exit from SPSS Modeler.
2. Edit the file *<modeler server installation path>/ext/bin/pasw.entityanalytics/ea.cfg* to increase the value of the following option:

```
timeout, value
```

where *value* is the timeout value in seconds for the entity analytics engine (default is 60).

To adjust the timeout for the entity analytics database server (solidDB only)

1. Exit from SPSS Modeler.
2. Edit the file `<modeler server installation path>/ext/bin/pasw.entityanalytics/ea.cfg` to increase the value of the following option:

`solid_timeout, value`

where *value* is the timeout value in seconds for the entity analytics solidDB database server (default is 100).

Running IBM SPSS Modeler Entity Analytics with SPSS Modeler client and SPSS Modeler Server on the same Windows system

If you have installed IBM SPSS Modeler Entity Analytics into both SPSS Modeler client and SPSS Modeler Server on the same Windows system, by default both client and server will share the same repository. If you want them to use separate repositories, you need to edit the configuration file *ea.cfg* on **one** of the systems to configure it to use a different port range and repository folder.

Note: In particular, if you use a 32-bit SPSS Modeler client and a 64-bit SPSS Modeler Server (or vice versa), you will need to carry out this procedure.

1. Open the file `<modeler [server] installation path>/ext/bin/pasw.entityanalytics/ea.cfg` for editing.
2. Change the `min_port` and `max_port` settings to use different ports from the other system. For more information, see the topic [Configuring port assignments](#) on p. 40.
3. Change the `repository_data_directory` setting to use a different directory from the other system.
4. Save and close the *ea.cfg* file.

Purging an entity repository

If you want to clear out the data records from an entity repository, but maintain the configuration information, use the repository purge option.

To purge a repository

1. Open an Entity Analytics node.
2. Click the Entity repository list.
3. Click `<Browse...>` to display the Entity Resolution Instances dialog box.
4. On the Entity Resolution Instances dialog box, click the Repository Name list.
5. Select the repository you want to purge.
6. If you are not already connected, enter the administrator username and password and click Connect.
7. When the Purge Repository button is enabled, click it.

8. Click Purge Repository on the Purge Data Sources dialog box to confirm purging of the repository.

Deleting an entity repository

When you no longer need a repository, you can delete it completely.

Caution: This does exactly what it says. **You cannot undo this operation.** If you are not sure, use the Purge button to remove all the source data. Doing so does not remove the repository configuration. For more information, see the topic [Purging an entity repository](#) on p. 43.

Note: The following procedure assumes that you can connect to the repository from SPSS Modeler, and that you know the administrator username and password for the database that hosts the repository. If this is not the case, follow the procedure for deleting a repository when unable to connect to it. For more information, see the topic [Deleting a repository when unable to connect to it](#) on p. 44.

To delete a repository

1. Open an Entity Analytics node.
2. Click the Entity repository list.
3. Click <Browse...> to display the Entity Resolution Instances dialog box.
4. On the Entity Resolution Instances dialog box, click the Repository Name list.
5. Select the repository you want to delete.
6. If you are not already connected, enter the administrator username and password and click Connect.
7. When the Delete Entire Repository button is enabled, click it.
8. Click Delete to confirm deletion of the repository.
9. Click OK to acknowledge successful deletion.

Deleting a repository when unable to connect to it

Use the following procedure if you want to delete an entity repository but are unable to connect to it, either because of connectivity issues with SPSS Modeler or because you have forgotten the username or password.

Perform this procedure at the machine that hosts the repository database.

Windows systems

1. Open a Command Prompt window.
2. Enter:

```
cd modeler_install_dir  
cd ext\bin\pasw.entityanalytics\tools
```

```
delete_repository.bat repos_name
```

where *modeler_install_dir* is the directory where SPSS Modeler is installed, and *repos_name* is the name of the repository.

3. Continue from “Completing the procedure” later in this section.

UNIX systems

1. Open a shell.
2. Enter:

```
cd modeler_server_install_dir  
cd ext/bin/pasw.entityanalytics/tools  
./delete_repository.sh repos_name
```

where *modeler_server_install_dir* is the directory where SPSS Modeler Server is installed, and *repos_name* is the name of the repository.

Completing the procedure (all systems)

1. At the prompt, confirm deletion of the repository by entering Y.
2. When the repository has been deleted, you see the message:

```
Info - please remove repository files from the following directory:  
    directory_path  
(Note - you may need to reboot first before you can remove the repository files)
```

3. Delete the directory that has the same name as the repository you deleted. If you are unable to delete the directory, restart the machine and try again.

Entity analytics in action

About this example

In this example, we'll see how adding entity analytics can improve still further on the already impressive results you can get from using IBM® SPSS® Modeler.

This example uses the stream *loan_entity_analytics.str*, which references the data file *loan_applications.csv*. These files are available from the *Demos* directory of any SPSS Modeler installation that also has IBM® SPSS® Modeler Entity Analytics installed. You can access the *Demos* directory from the SPSS Modeler program group on the Windows Start menu. The *loan_entity_analytics.str* file is in the *Entity_Analytics* directory.

Note: Before you can run this example stream, you need to create a repository on your system. Do so before continuing with this example. For more information, see the topic [Creating the repository](#) in Chapter 3 on p. 16.

Let's start with a familiar situation – executives of a bank are concerned about whether customers are likely to default on loans for which applications are pending. The IT department of the bank is a long-time user of SPSS Modeler, so their staff have already created a stream and built a predictive model from their existing data about 700 loans the bank has made in the past. These loans have either been repaid, or customers have defaulted on their repayments.

The original model

Here's how the bank staff built their model and what they learned from it.

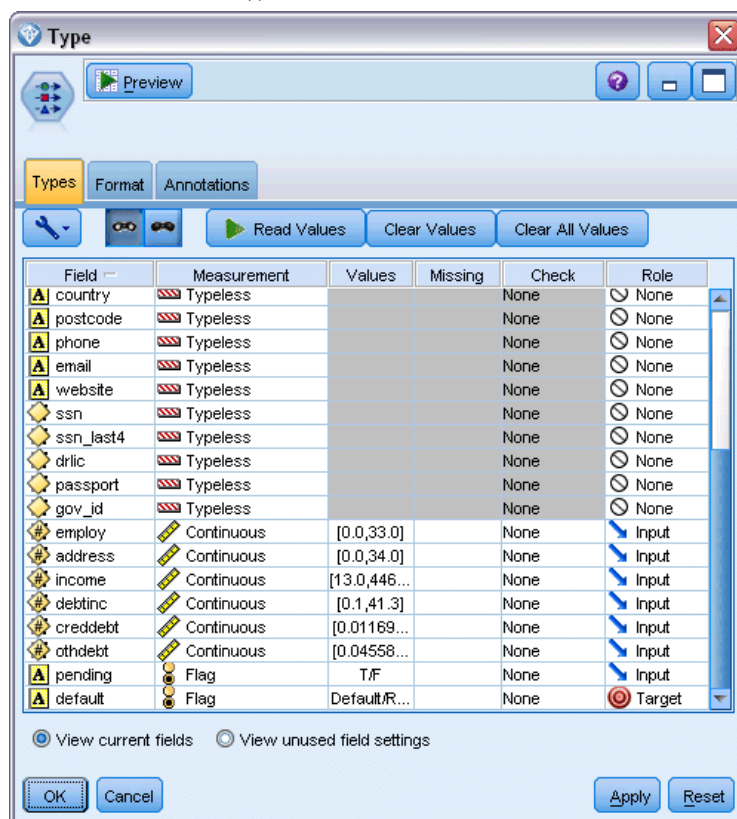
Figure 4-1
Initial stream with modeling node



As well as details of past loans, the *loan_applications.csv* data set includes details of 150 customers whose loan applications are still pending, giving a total of 850 records.

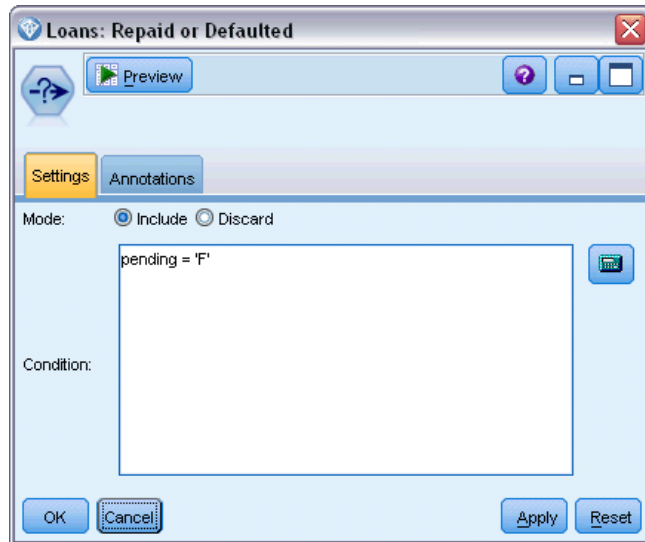
Not all of the fields from the data set are useful in making the prediction—for example, name fields can be ignored. The *Type* node filters out the fields to ignore by setting their role to None. Fields to be used to make the prediction have their role set to Input, and the field whose value the model is trying to predict has its role set to Target.

Figure 4-2
Field roles set in the Type node



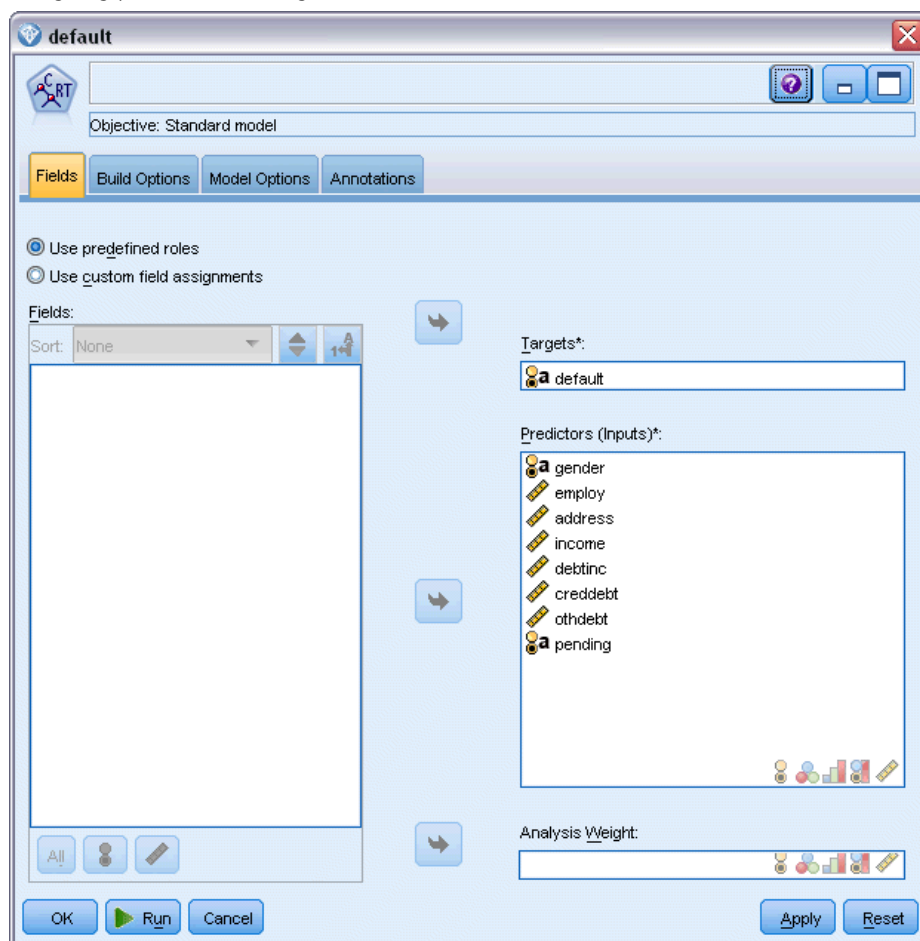
As the model must make its predictions based solely on the past data, the stream includes a Select node that includes only those loans that are *not* marked as Pending, thus discarding the 150 pending loans.

Figure 4-3
Discarding the pending loan applications



With the pending loans discarded, only the remaining 700 loans that were either repaid or defaulted have their details passed to the modeling node. The bank could have used one of a number of SPSS Modeler algorithms to produce a good model. In this case, they have used a C&R Tree node, which will be used to build a model to predict likely defaulters based on the past performance of the bank's customers.

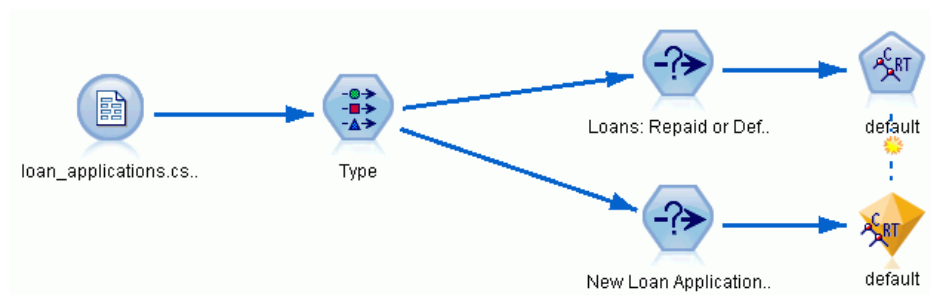
Figure 4-4
Assigning predictor and target fields



The fields used to make the prediction are designated as the predictor fields, and the field whose value the model is trying to predict—default in this case—is set as the target field, as defined earlier by the Type node.

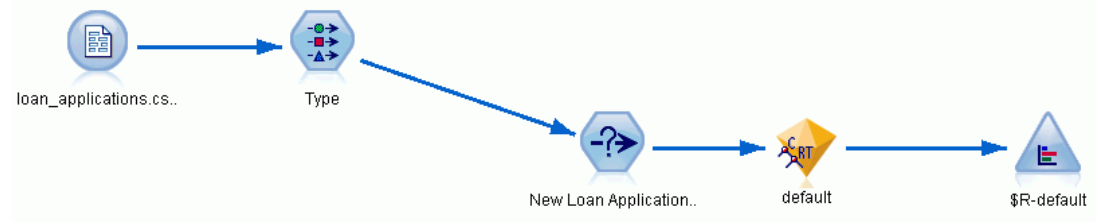
Running this stream produces a model nugget, containing the model that has been built from the predictor fields.

Figure 4-5
Stream with model nugget added



Now the bank's analyst can use the model to start predicting whether customers with pending repayments are likely to default. Using the original data set, the analyst inserts a Select node that this time includes only the 150 loan records marked as Pending, instead of discarding them. The analyst passes these records directly into the model, adding a Distribution node for a visual representation of the model's predictions.

Figure 4-6
Stream selecting new loan applications and with Distribution node added



The Distribution node shows the distribution of the values of the *\$R-default* field in the model. This field is added to the data model by the C&R Tree node when it is run. The field contains the prediction of whether each new applicant will repay or default, and we'll be using this field later to compare the effect of adding entity analytics.

Running this part of the stream, the analyst learns from the output of the Distribution node that 137 of the 150 new applicants are predicted to repay their loans. The remaining 13 are predicted to default, so the analyst will probably recommend the bank to reject their applications.

Figure 4-7
Output from the Distribution node without entity analytics

Value	Proportion	%	Count
Default		8.67	13
Repaid		91.33	137

Adding entity analytics

Now let's see if the situation can be improved by adding entity analytics into the equation. Imagine that you are an entity analytics specialist, called in by the bank to investigate possible fraudulent entries in the customer records in the source data. There might be duplicate records caused by data entry errors, but it's also possible that a loan applicant might be trying to disguise their identity. In any case, the bank needs to know the true picture.

For the purposes of this example, we'll assume that an entity repository has already been created. For more information, see the topic [Creating the repository](#) in Chapter 3 on p. 16.

Getting the source data into the repository

First, you need to add an EA Export node to the data source node so that you can export the source data into the entity repository.

Figure 4-8

Attaching an EA Export node to the data source node



Before you can export the data, you need to map fields in the data source to features in the entity repository. This is necessary because different data sources may use different field names for the same type of information. The entity repository provides a standard set of information types (known as “features”) to avoid duplication.

Figure 4-9

Mapping fields to features

The screenshot shows the 'Entity Analytics Export' dialog box. It has a 'Refresh' button and a 'Repository' tab selected. The 'Mode' is set to 'Add to repository'. The 'Entity Repository' is 'dba@REPOS1', 'Source tag' is 'TEST', and 'Entity type' is 'PERSON'. The 'Unique Key' is 'key'. A table below shows the mapping of source fields to repository features.

Field	Mapped to Feature	Usage
key		(Auto)
fname	NAME.GIVEN_NAME	(Auto)
mname	NAME.MIDDLE_NAME	(Auto)
lname	NAME.SUR_NAME	(Auto)
generation	NAME.NAME_GEN	(Auto)
dob	DOB.DOB	(Auto)
dod		(Auto)
gender	GENDER.GENDER	(Auto)
addr1	ADDRESS.ADDR1	(Auto)

Buttons at the bottom include 'Import Mapping...', 'Export Mapping...', 'OK', 'Run', 'Cancel', 'Apply', and 'Reset'.

In the EA Export node, set up the details about the repository: the connection details, the source tag (to identify the data source – TEST in this case), the entity type (the set of features we’re using – the one named PERSON), and the unique key field (to uniquely identify each record). In this case, use the key field as the unique key.

Now you can set up the mappings. In the feature set you're using, there are features corresponding to the fields *fname*, *mname*, *lname*, *generation*, *dob*, *gender*, *addr1*, *city*, *country*, *postcode*, *phone*, *email*, *ssn*, *drlic*, and *passport*.

Start by setting up the mapping for *fname*. Double-click the Mapped to Feature column in the table on the *fname* row, scroll down to the NAME.GIVEN_NAME entry, and click it to create the mapping.

Now map the remaining fields that have corresponding features, so that the full set of mappings looks like this.

Table 4-1
Fields mapped to repository features

Field	Mapped to Feature
<i>fname</i>	NAME.GIVEN_NAME
<i>mname</i>	NAME.MIDDLE_NAME
<i>lname</i>	NAME.SUR_NAME
<i>generation</i>	NAME.NAME_GEN
<i>dob</i>	DOB.DOB
<i>gender</i>	GENDER.GENDER
<i>addr1</i>	ADDRESS.ADDR1
<i>city</i>	ADDRESS.CITY
<i>country</i>	ADDRESS.COUNTRY
<i>postcode</i>	ADDRESS.POSTAL_CODE
<i>phone</i>	PHONE.PHONE_NUM
<i>email</i>	EMAIL_ADDR.ADDR
<i>ssn</i>	SSN.ID_NUM
<i>drlic</i>	DRLIC.ID_NUM
<i>passport</i>	PASSPORT.ID_NUM

Click Run to export the data into the repository. This takes a little while, so when the Execution Feedback dialog box closes, the export is complete.

Reading the resolved identities

As you export the data to the repository, the entity analytics system starts resolving possible identity conflicts, assigning a unique entity identifier, which you'll see later as the *\$EA-ID* field. (*Note:* This is not the same as the Unique Key field in the EA Export node—that field is used to uniquely identify data source records.)

The first step to reading the resolved identities is to add an Entity Analytics(EA) source node to the stream. This source node should not be connected to anything at this stage.

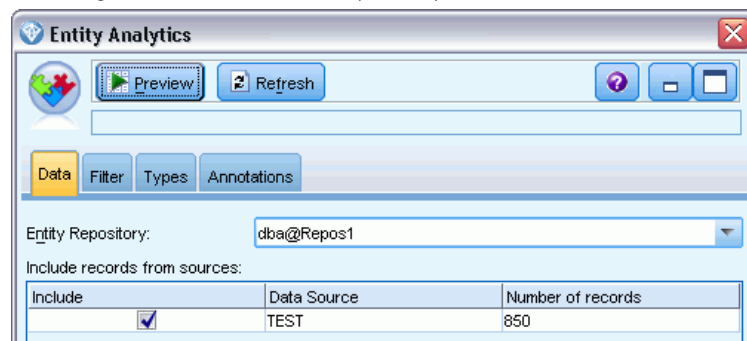
Figure 4-10
Entity Analytics(EA) source node



Entity Analytics(EA)

Open the Entity Analytics(EA) source node and set the Entity Repository details. A list is then displayed of data sources that have been exported to the repository– in this case there is only one.

Figure 4-11
Selecting a data source in the repository



Select the check box for the TEST data source and click OK.

Let's take a look at what the entity analytics system has done to the data. Attach a Table node to the Entity Analytics(EA) source node, open the Table node and click Run to display the Table node output window.

Figure 4-12
Output from the Table node

	\$EA-ID	\$EA-SRC	\$EA-KEY
1	1	TEST	REF00000001
2	2	TEST	REF00000002
3	3	TEST	REF00000003
4	4	TEST	REF00000004
5	5	TEST	REF00000005
6	6	TEST	REF00000006
7	7	TEST	REF00000007
8	8	TEST	REF00000008
9	9	TEST	REF00000009
10	10	TEST	REF00000010
11	11	TEST	REF00000011
12	12	TEST	REF00000012
13	13	TEST	REF00000013
14	14	TEST	REF00000014
15	15	TEST	REF00000015
16	16	TEST	REF00000016
17	17	TEST	REF00000017
18	18	TEST	REF00000018
19	19	TEST	REF00000019
20	20	TEST	REF00000020

Only one field looks familiar – the one labeled *\$EA-KEY*. This is actually the *key* field from the source data, and it's here because you chose it as the Unique Key field in the EA Export node.

The system has added two other fields though. The *\$EA-ID* field is the unique identifier, not of the source records, but of the resolved identities. We'll see the difference in a moment. The *\$EA-SRC* field identifies where the data came from – it says TEST here because that's the source tag that you assigned it in the EA Export node.

What's happened to all the other fields in the source data? Don't worry, they're still in the repository – it's just that, for performance reasons, the Entity Analytics(EA) source node passes only the minimum set of fields downstream for further processing.

Now, scroll the Table node output down to row 385.

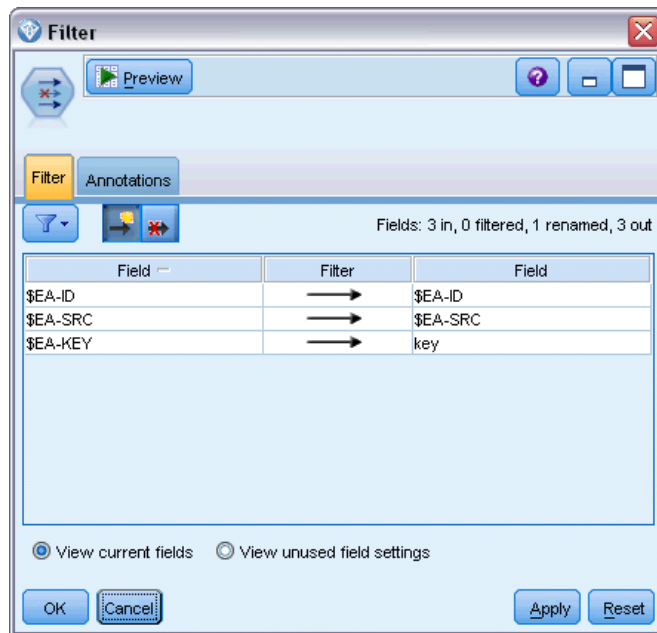
Figure 4-13
Differences between Table output rows and *\$EA-ID* numbers

	<i>\$EA-ID</i>	<i>\$EA-SRC</i>	<i>\$EA-KEY</i>
376	376	TEST	REF00000376
377	377	TEST	REF00000377
378	378	TEST	REF00000378
379	379	TEST	REF00000379
380	380	TEST	REF00000380
381	381	TEST	REF00000381
382	382	TEST	REF00000382
383	383	TEST	REF00000383
384	384	TEST	REF00000384
385	45	TEST	REF00000385
386	386	TEST	REF00000386
387	387	TEST	REF00000387
388	388	TEST	REF00000388
389	389	TEST	REF00000389
390	390	TEST	REF00000390
391	391	TEST	REF00000861
392	392	TEST	REF00000392
393	393	TEST	REF00000393
394	394	TEST	REF00000394
395	395	TEST	REF00000395
396	396	TEST	REF00000396

Notice how the *\$EA-ID* number appears to be out of sequence here. The entity analytics system has determined that record REF00000385 references the person identified as entity 45, who also has the record REF00000045. Scrolling further down the output, there are more numbers out of sequence, for example at rows 485, 517, 520 and so on. We'd better take a closer look.

First, let's highlight the fact that the *\$EA-KEY* field contains the data from the *key* field in the source data, by renaming it to *key*. Attach a Filter node to the Entity Analytics(EA) source node and open the Filter node. Double-click the string *\$EA-KEY* in the second Field column and type *key*.

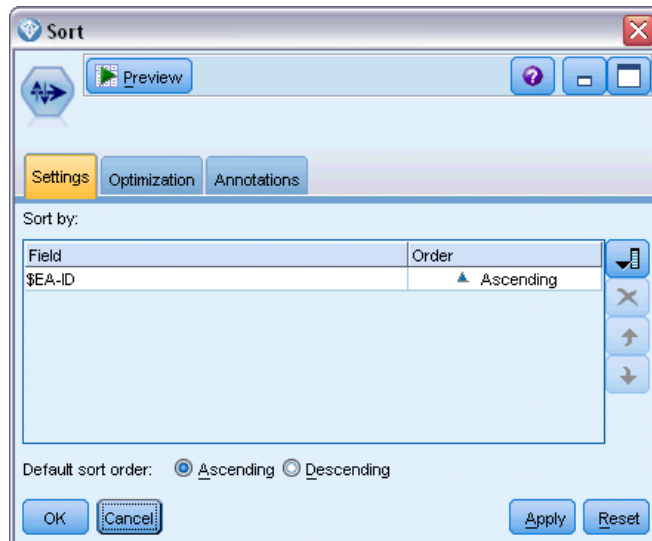
Figure 4-14
Renaming the \$EA-KEY field



Click OK to close the Filter node.

Now we need to sort the *\$EA-ID* entity IDs into ascending order. Attach a Sort node to the Filter node. Open the Sort node, click the top button next to the Sort By table, select *\$EA-ID* and click OK.

Figure 4-15
Sorting the entity IDs into ascending order



Leave the sort order as Ascending and click OK.

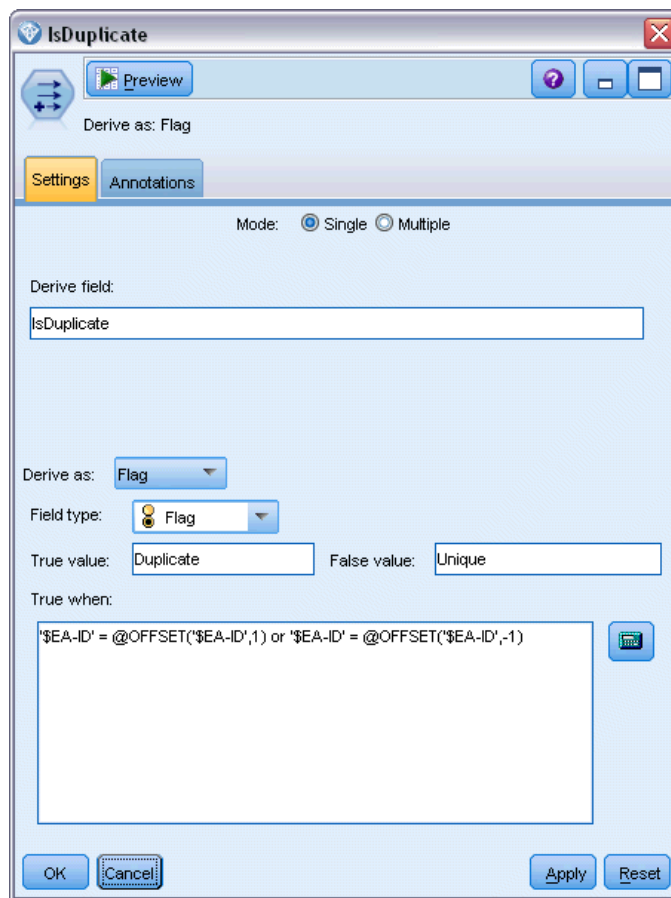
Now you need to create an extra field that indicates whether a record is unique or a duplicate. Attach a Derive node to the Sort node. Open the Derive node and set the Derive field name to *IsDuplicate*. From the Derive as list, choose Flag, which also sets the Field type to Flag. Set the True value field to Duplicate and the False value field to Unique.

To find duplicate records, you'll make use of a special sequence function, named `@OFFSET`, which is provided with SPSS Modeler.

Type the following in the If field:

```
'$EA-ID' = @OFFSET('$EA-ID',1) or '$EA-ID' = @OFFSET('$EA-ID',-1)
```

Figure 4-16
Setting the condition in the Derive node



With the entity IDs sorted into ascending order, the `@OFFSET` function checks whether adjacent entity IDs are identical, in which case the records are duplicates. If so, their *IsDuplicate* value is set to *Duplicate*, otherwise it is set to *Unique*.

Click OK to close the node.

To see the effect of the Derive node, attach a Table node to the Derive node, open the Table node and click Run. Scroll the Table node output window down to row 45.

Figure 4-17
Output from the Derive node

	\$EA-ID	\$EA-SRC	key	IsDuplicate
39	39	TEST	REF00000039	Unique
40	40	TEST	REF00000040	Unique
41	41	TEST	REF00000041	Unique
42	42	TEST	REF00000042	Unique
43	43	TEST	REF00000043	Unique
44	44	TEST	REF00000044	Unique
45	45	TEST	REF00000045	Duplicate
46	45	TEST	REF00000385	Duplicate
47	46	TEST	REF00000046	Unique
48	47	TEST	REF00000047	Unique
49	48	TEST	REF00000048	Unique
50	49	TEST	REF00000049	Unique
51	50	TEST	REF00000050	Unique
52	51	TEST	REF00000051	Unique
53	52	TEST	REF00000052	Unique
54	53	TEST	REF00000053	Unique
55	54	TEST	REF00000054	Unique
56	55	TEST	REF00000055	Unique
57	56	TEST	REF00000056	Unique
58	57	TEST	REF00000057	Unique

Remember when we looked at the output directly from the Entity Analytics(EA) source node. The system had already identified that that record REF00000385 references the same person as entity 45. Now we've taken this one stage further and flagged the fact that records REF00000045 and REF00000385 are duplicates, as they both reference entity 45.

Scroll the output window down further and you'll see the other records that are flagged as duplicates.

To obtain a report listing the duplicate records, attach a Report node (from the Output tab of the nodes palette) to the *IsDuplicate* Derive node. Open the Report node, copy the following text into the input field of the Template tab, and click Run.

```
<html>
<h1>List of duplicate customer records.

<h2>This report was generated: [@TODAY]

<h2>Duplicate records
<table>
<tr>
  <td>Entity ID</td>
  <td>Key</td>
</tr>

#WHERE IsDuplicate = "Duplicate"
```

```

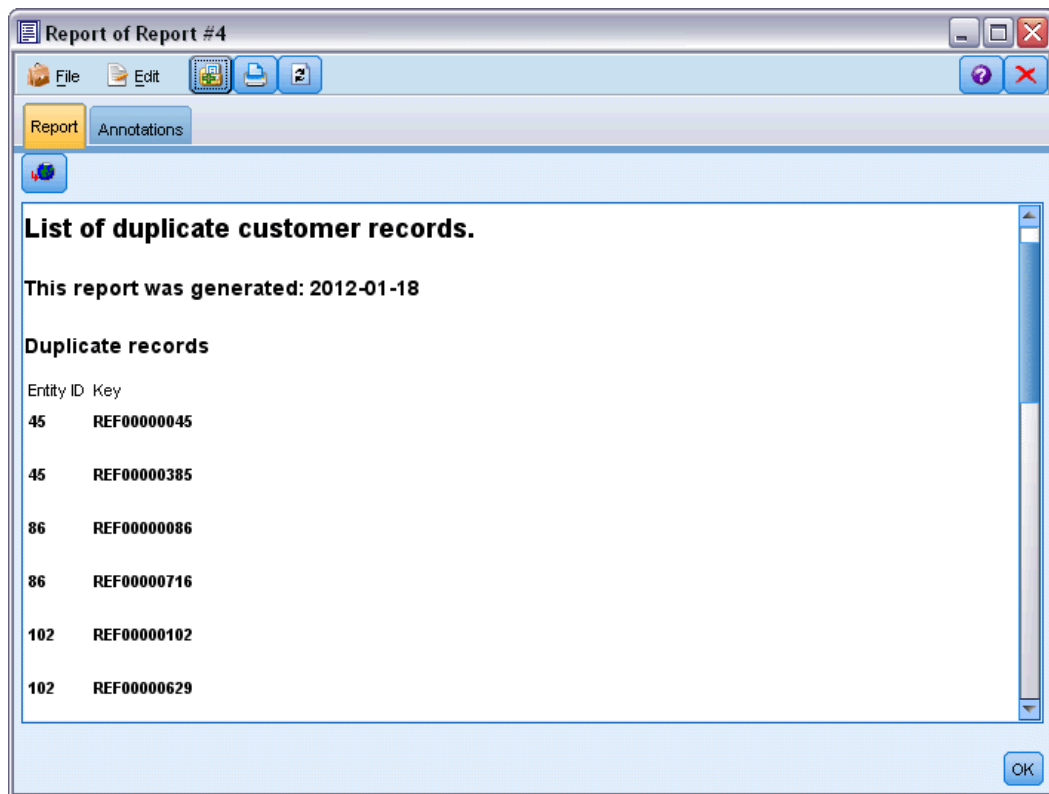
<tr>
  <td>['$EA-ID']</td>
  <td>[key]</td>
</tr>
#
</table>

</html>

```

This gives the following output.

Figure 4-18
Output from the Report node



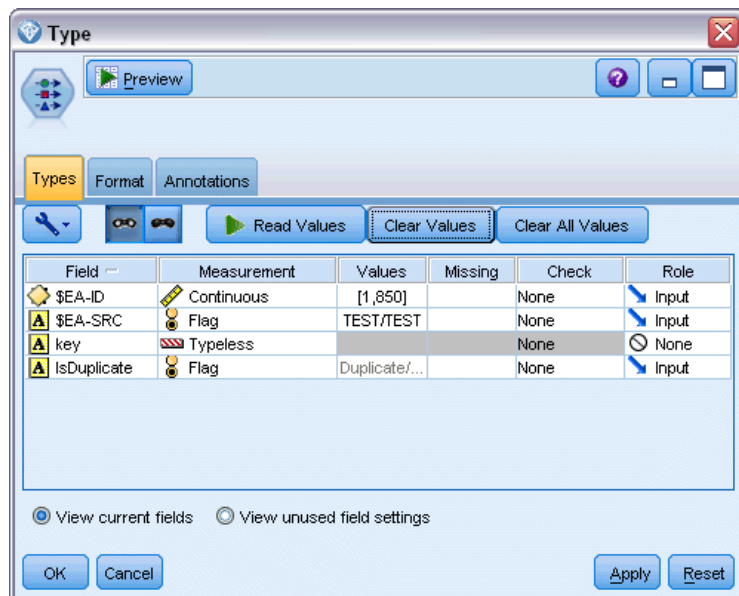
The report makes use of HTML format in this case, though you could also use XML or ASCII formatting.

Comparing entity analytics output with the original model

The final stage of this example is to see whether adding entity analytics makes any difference to the bank's original prediction. You may remember that the original model predicted 13 defaulters out of the 150 pending applications. You're going to use a Merge node to merge the output from that model with information about duplicate records from entity analytics to see if doing so changes the prediction.

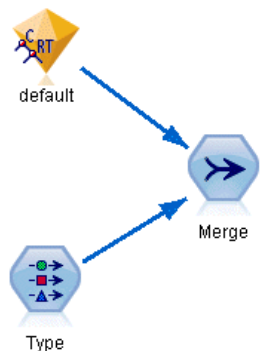
First, you need to ensure that the new fields added by entity analytics have the correct data types, or *measurement levels* as they are known in SPSS Modeler. Attach a Type node to the IsDuplicate Derive node, open the Type node and click the Read Values button.

Figure 4-19
Settings for the Type node



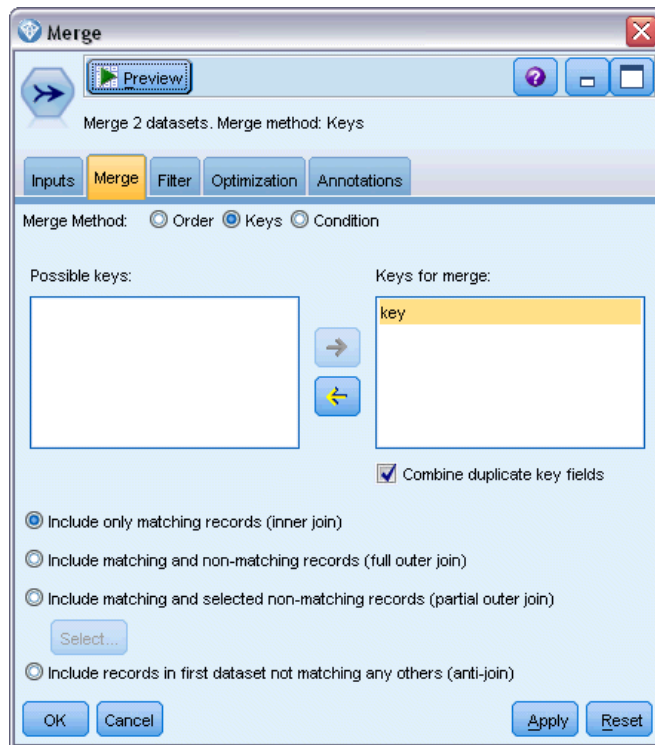
Now you can add the Merge node. Attach it to the Type node, and also connect it to the gold nugget containing the original model. To do so, right-click the gold nugget, choose Connect and then click the Merge node, which should now have two input arrows.

Figure 4-20
Inputs to the Merge node



Open the Merge node, set the Merge Method to Keys, and click the right-arrow button to move the key field from Possible keys to Keys for Merge, then click OK.

Figure 4-21
Specifying the key field for the merge operation



You're almost ready to make the comparison now. However, if you were to attach a Distribution node and run it at this point, you wouldn't see any change from the original prediction. Although the stream now merges the output from the original model nugget with the new fields created by entity analytics, the prediction field itself (*\$R-default*) in the data model hasn't been updated with the new information.

To do this, you'll use a Filler node, which can replace field values. Attach a Filler node to the Merge node and open the Filler node.

Click the top button to the right of Fill in fields, scroll towards the bottom of the list, choose *\$R-default* and click OK. This is the field whose values are to be changed if the condition specified in the rest of the dialog box is fulfilled.

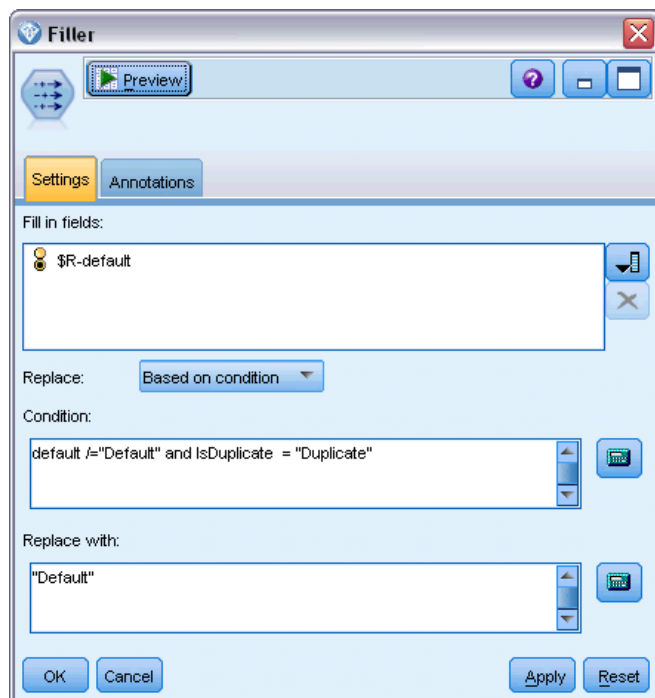
To specify the condition, ensure that Replace is set to Based on condition, then in the Condition field enter:

```
default != "default" and IsDuplicate = "Duplicate"
```

In the Replace with field, enter:

```
"default"
```

Figure 4-22
Specifying the condition for replacing field values



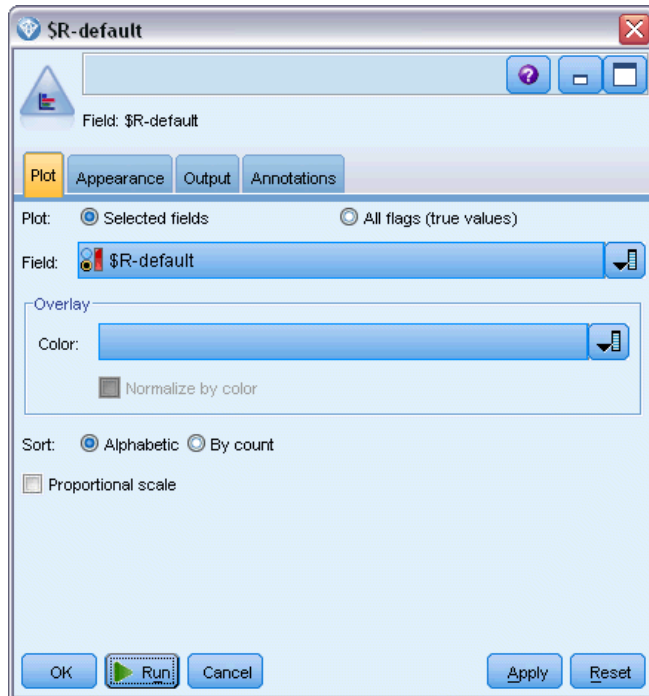
These settings need a little explanation. The condition states that, for each record where the value of the *default* field from the original data set is not equal to Default and the record has been flagged as a duplicate, then the value of the *\$R-default* field in the model is set to Default.

The *\$R-default* field is the field in the model that contains the prediction of whether a customer is likely to default on the loan. In this way, customers with duplicate records are added to the model as potential defaulters.

Click OK to close the Filler node.

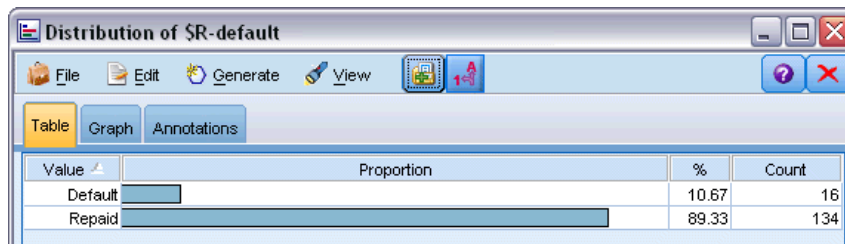
You're finally ready to see the difference that entity analytics has made. From the Graphs palette, attach a Distribution node to the Filler node and open the Distribution node. Click the Field list and choose *\$R-default*.

Figure 4-23
Settings for the Distribution node



Click Run to produce the chart of the new prediction.

Figure 4-24
Output from the Distribution node after entity analytics



Now there are 16 risky applications instead of 13. These extra ones could be very costly if they did default, so you can graphically demonstrate to the bank the benefit of adding entity analytics to their risk assessment operations.

Summary

This example has shown how, by using entity analytics, you can eliminate duplication of records in data about people or organizations, leading to improved prediction quality.

Note: Ideally, you would eliminate duplicate records before doing any other processing. You could follow this up by using an Automated Data Preparation (ADP) node to analyze your data and identify fixes, screen out fields that are problematic or not likely to be useful, derive new attributes when appropriate, and improve performance through intelligent screening techniques.

Combining entity analytics and automated data preparation ensures that you are working with data that is as clean as possible.

Scripting Properties for IBM SPSS Modeler Entity Analytics

Scripting in IBM® SPSS® Modeler Entity Analytics is a powerful tool for automating processes in the user interface. Scripts can perform the same types of actions that you perform with a mouse or a keyboard, and you can use them to automate tasks that would be highly repetitive or time consuming to perform manually. For an explanation of using scripting, see the *ScriptingAutomation.pdf* guide available with IBM® SPSS® Modeler.

Common properties

Properties that are common to IBM SPSS Modeler Entity Analytics nodes are listed in the following table. Information on specific nodes is given in the sections that follow.

Property name	Data type	Property description
entity_repository	[<i>'field'</i> , <i>'field'</i> , ..., <i>'field'</i>]	The repository connection string. Format: [<i>'reposname'</i> , <i>'username'</i> , <i>'password'</i>] Example: entity_repository = ['repos1', 'dba', 'psw1']
entity_type	<i>string</i>	The entity type (set of features) to be used. Example: entity_type = 'PERSON'

entityanalytics_exportnode properties



The EA Export node is a terminal node that reads entity data from a data source and exports the data to a repository for the purpose of entity resolution.

entityanalytics_exportnode properties	Data type	Property description
mode	Add PurgeFirst	Export mode. Add adds source file records to existing contents of repository; PurgeFirst removes existing contents before exporting.
source_tag	<i>string</i>	The data source identifier. Example: source_tag = 'CUST'

entityanalytics_exportnode properties	Data type	Property description
unique_key_field	<i>string</i>	Input field to use for unique identifiers for data records. Example: unique_key_field = 'ID'
field_mapping	[['field_name' 'feature_element' 'usage_type']...]	Maps input fields to corresponding feature in repository. Example: field_mapping = [['fname' 'NAME.GIVEN_NAME' ''] ['addr1' 'ADDRESS.ADDR1' 'PRIMARY']] <i>Note:</i> To set <i>usage_type</i> to the equivalent of "(Auto)", use "" as in the first example above.

entityanalytics_sourcenode properties



The Entity Analytics(EA) source node reads the resolved entities from the repository and passes this data to the stream for further processing, such as formatting into a report.

entityanalytics_sourcenode properties	Data type	Property description
source_tags	<i>list</i>	List of tags for data sources that have been input to the repository. Example: source_tags=['LOANS', 'CUSTOMERS']

entityanalytics_processnode properties



The Streaming EA node compares new cases against the entity data in the repository.

entityanalytics_processnode properties	Data type	Property description
match	Exact ByIdentifier All	Matching criterion for retrieving entities from repository. Exact returns only exact matches. ByIdentifier returns exact matches and entities sharing same identifiers. All returns all possible matches.

Notices

This information was developed for products and services offered worldwide.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

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

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing, Legal and Intellectual Property Law, IBM Japan Ltd., 1623-14, Shimotsuruma, Yamato-shi, Kanagawa 242-8502 Japan.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Software Group, Attention: Licensing, 233 S. Wacker Dr., Chicago, IL 60606, USA.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, and SPSS are trademarks of IBM Corporation, registered in many jurisdictions worldwide. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other product and service names might be trademarks of IBM or other companies.



Index

- administrator credentials
 - managing for entity analytics, 41
- configuration
 - entity repository, 22, 31–32
- data source, selecting for entity analytics, 32
- data sources
 - connecting with entity analytics, 5, 15
 - viewing for entity analytics, 18, 38
- deleting
 - entity repository, 44
- EA Export node, entity analytics, 14
- entity analytics
 - compared with predictive analytics, 2
 - defined, 1
 - using with IBM SPSS Modeler, 4
 - using with other IBM SPSS products, 39
- Entity Analytics(EA) source node, 32
- entity matching, setting threshold, 30
- entity repository, 14
 - administrative tasks, 39
 - comparing new cases against, 34
 - configuring, 22, 31–32
 - configuring port assignments, 40
 - connecting to IBM SPSS Modeler, 7
 - creating, 6, 16–17
 - deleting, 44
 - features, 26
 - maintaining, 24
 - managing administrator credentials, 41
 - moving to a different storage directory, 41
 - options, 18
 - purging, 43
 - setting stream properties, 42
 - setting up, 14
- entity types
 - entity analytics, 28
 - entity repository, 20
- export nodes
 - entity analytics, 8, 14
- exporting
 - data to an entity repository, 8
- features
 - entity repository, 7, 19–21, 23–24, 26, 35, 37–38
- identity resolution, entity analytics, 8
- legal notices, 66
- mapping fields
 - to entity repository features, 7, 19–21, 23, 35, 37–38
- new cases, comparing against entity analytics repository, 34
- nodes
 - adding to an entity analytics stream, 34
- output
 - from entity analytics, 39
- port assignments
 - configuring for entity analytics, 40
- process nodes
 - entity analytics, 11, 34
- properties
 - scripting, 64
- purging
 - entity repository, 43
- renaming
 - data fields for entity analytics, 33
- repository
 - administering entity analytics, 39
 - entity analytics, 6–8, 14, 16–24, 26, 28, 31–32, 34–35, 37–38, 43–44
 - storage directory for entity analytics, changing, 41
 - resolution rules, entity analytics, 30
 - resolved identities, analyzing with entity analytics, 32
 - resolving identities, entity analytics, 8
- scripting
 - properties, 64
- source nodes
 - entity analytics, 11, 32
- source tags
 - entity repository, 20
- stream properties
 - setting for entity analytics, 42
- Streaming EA node, entity analytics, 34
- threshold for entity matching, entity analytics, 30
- trademarks, 67
- type information, setting for entity analytics, 33
- unique keys
 - entity analytics, 7
 - entity repository, 20
- usage types, entity analytics, 28