



USER MANUAL

VERSION 2012-05-02



Commetrix Producer

Producing CMX Files from File Templates

Product Website: www.commetrix.net
Vendor Contact : info@trilexis.com

CSV

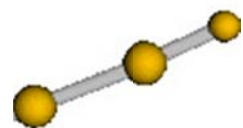
➔

XLS

➔

XML

➔



.cmx for Commetrix Analyzer

Contents

| | | |
|-----|---|----|
| 1 | Introduction | 4 |
| 2 | Licensing Producer | 4 |
| 3 | Installation of Commetrix Producer | 4 |
| 4 | Logging into the Commetrix Producer | 5 |
| 5 | Importing Raw Data from Tables..... | 6 |
| 5.1 | Importing from .CSV Tables..... | 6 |
| 5.2 | Using the Excel Template for Import..... | 7 |
| 6 | The Tables of the Commetrix Model (.csv & .xls) | 9 |
| 6.1 | Node Table | 9 |
| 6.2 | Linkevent Table | 11 |
| 6.3 | Linkeventsender Table | 12 |
| 6.4 | Linkeventrecipient Table | 13 |
| 6.5 | Linkeventparent Table | 13 |
| 7 | Defining Import Parameters | 14 |
| 7.1 | Network Parameters | 14 |
| 7.2 | Node Parameters | 15 |
| 8 | Starting the Production Process..... | 16 |
| 9 | Contact | 16 |

1 Introduction

COMMETRIX is a software framework for the analysis of social networks. Next to conventional static network analysis of cumulative snapshots of relationship networks, its event-based network model enables the examination of network evolution over time as well as the study of link and node properties.

Commetrix offers an innovative **DATA MODEL**. Unlike most other SNA tools, it does not store links directly as valued relationships but as individual relational events. This provides the opportunity of modelling networking processes with multiple types of relationships and comprehensive qualitative and quantitative node and link attributes in a single dataset. With this approach, users can encode aspects like topic descriptors (e.g. content coding, keywords), types of links, e.g. socialization, document exchange, affiliation, media, time stamps of links, links connecting multiple nodes, as well as arbitrary quantitative or qualitative variables classifying nodes, e.g. affiliation, age, types, etc.

The **USER INTERFACE** is very easy to use. It is developed to visually support exploratory examination of a network dataset in order to identify and observe relevant substructures, periods, and processes of your network data. Commetrix computes time window measures and additionally provides very sophisticated functionality for displaying and animating the community evolution as an evolving graph – allowing the user to visually inspect the actors' activities. The animated graph, called *communigraph*, is one of the best existing visualizations of network change. Visual variables can be set by the user to represent node and link properties by label, node size, node color (brightness, transparency), or a number of rings around the node.

Complex options for time, actor, relationship, and topic filtering help to **FOCUS RELEVANT STRUCTURES**, i.e. relevant nodes, relationships, time periods, or even topics. For any time period and for any selection, typical social network measures can be computed and analyzed. Selections can be exported to tables for further analysis. This includes the export of network changes over time. All these features help to actually represent and visually trace change in a network and adds additional insight to the quantitative results.

This manual gives an introduction to the Commetrix Producer which retrieves network data from Excel and other file formats and creates Commetrix network files (.cmx) that can be visualized by the Commetrix Analyzer.

Also note the online video introduction to producing network files with Commetrix Producer:

<http://www.youtube.com/watch?v=CxpUHrf6DNc>

2 Licensing Producer

Before you acquire a license of Commetrix Producer (also referred to as CMXProducer) you can try it with smaller datasets for free (see section 4) via the [trial version](#).

If you want the [full license](#), please have a look at: <http://www.commetrix.net/lic> for documents explaining possible Commetrix-based solutions and for information about the ordering process.

3 Installation of Commetrix Producer

Installation is simply done by unpacking the downloaded .zip file into a folder of your choice (we would advise not to use the location C:/Program Files, as special protections may later prevent you from storing files or changing the folder's contents). We do not guarantee the CMXProducer working properly with other operating systems than Windows XP, Windows Vista and Windows 7.

After the installation you can either use the trial mode (see section 4) or acquire a license (see section 2). In the following, further software requirements (i.e. Java) for running the CMXProducer are explained:

To run the CMXProducer (as well as the CMXAnalyzer) a Java Virtual Machine is required. On most machines it is already installed. If not, download the Java Virtual Machine (<http://www.java.com/de/download/>) and follow the installation instructions given there. CMXProducer requires at least Java 1.6 (Java Version 6). We do not guarantee the CMXProducer working properly with older versions. Keep the path to the Java Runtime Environment (JRE) folder (<path-to-jre>) as it may be required later.

Commetrix Producer further uses two required external java libraries in compiled form, derby.jar and jxl.jar (generally, they require no extra installation):

Java Excel API, Version 2.6.12, (<http://www.jexcelapi.org/>), License: Lesser General Public License (LGPL), (<http://www.gnu.org/copyleft/lesser.html>)

Apache Derby, Version 10.5, (<http://db.apache.org/derby/>), License: Apache License, Version 2.0, (<http://www.apache.org/licenses/LICENSE-2.0.html>)

4 Logging into the Commetrix Producer

To run Commetrix Producer, either a valid license file together with username and password is required , or the trial mode can be selected.

The trial mode requires no licensing/activation process and there is no limited trial period. Rather, in the trial mode Commetrix Producer considers all nodes of the provided dataset but only the first 50 link events (e.g. messages) among them. That means that smaller datasets can be produced completely while larger files will yield an incomplete network datafile (.cmx), which can only be used to test functionality of Commetrix Producer or Commetrix Analyzer but not for a reliable analysis.

If you decide to license the Commetrix framework, then have a look at section 2 for further guidance. After the licensing process is carried out you receive a license activation file (License_Producer.dat), a login and a password (which you can propose).

After the username and password to run the Commetrix Producer have been entered (see figure below) and checked, the main interface is presented. The next section discusses its main functions.



Figure 1: Commetrix Producer – Login Window. The Trial mode is activated with the “Trial” button. The “Order” Button leads you to web-based information about solutions, prices, and about the ordering processes.

5 Importing Raw Data from Tables

The Commetrix Producer interface is shown in the figure below. The current version allows you to produce a .cmx network file from the following different raw data formats:

- Microsoft Excel
- Comma Separated Value Text Files (.csv) – a standard table format.
- XML



Figure 2: Commetrix Producer – User Interface.

Select the desired data sources by selecting either the ‘CSV to CMX’ tab, the ‘XLS to CMX’ tab, or the ‘XML to CMX’ tab (beta). The default and most used option is import from comma separated value tables.

5.1 Importing from .CSV Tables

If you select the tab ‘CSV to CMX’ (importing from standardized comma separated value files) a tab opens with five file selection boxes. You need to select an appropriate file for each of these boxes from your local data. It needs to follow the Commetrix .csv format (see below) Note that all five files together comprise the dynamic network model that is to be produced as a .cmx Commetrix file. Also define the used separator (‘;’ or ‘,’) in the .csv files (usually depending on national standards).

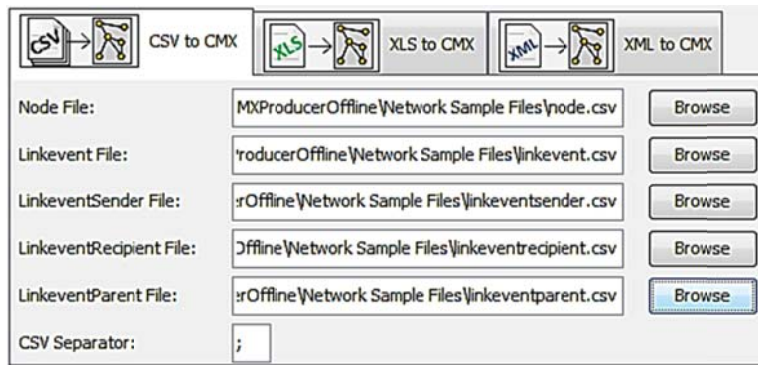


Figure 3: Commetrix Producer – .CSV File selection. Also define the used separator (‘;’ or ‘,’)

5.2 Using the Excel Template for Import

If you select the tab ‘XLS to CMX’ (importing from Microsoft Excel files) a tab opens with a file selection box. You need to select an appropriate .xls file from your local drives. It needs to follow the Commetrix .xls format (see below). Note that all five worksheets in the Excel table together comprise the dynamic network model that is to be produced as a .cmx Commetrix file.

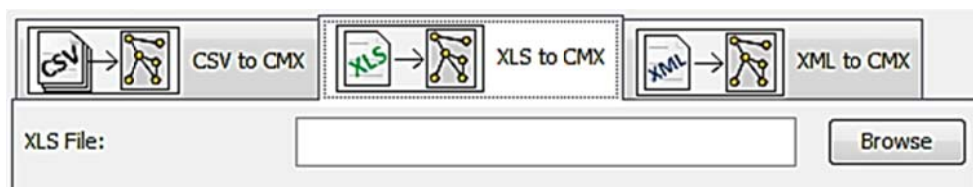


Figure 4: Commetrix Producer – .XLS File selection.

If we open the file CMX_Sample.xls as an example of Commetrix’s standard .xls format, we see the layout shown in the following figure.

There are five worksheets (see the bottom list of tabs in the Figure), Node, Linkevent, Linkeventsender, Linkeventrecipient and Linkeventparent. They exactly refer to the five .csv tables that are an alternative way of coding Commetrix input data (see sections 5.1 and 5.3).

Each worksheet begins with a header (see the colored cells ‘nodeID’, etc.) followed by the actual network data.

From the excel template, you can also export .csv tables to be used with Commetrix Producer.

The individual worksheets and their contents are described in the next section. Note that there is no difference between the .csv file standard and the .xls standard requirements for each worksheet in the Excel file.

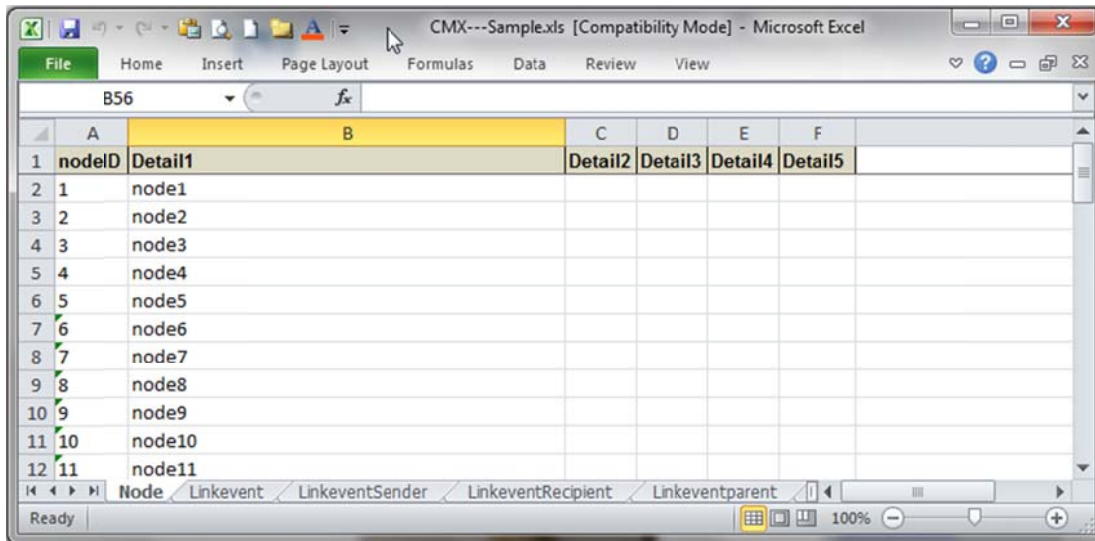


Figure 4: Commetrix .xls Table Template, opened in Excel.

5.3 Importing from XML File

Commetrix Producer allows importing from an XML File that is compliant with the Commetrix XML format description laid out in the specification: Commetrix_Schema.xml. A very simple example XML file is available in the provided demo data that comes with the download.

Just as with Excel files, the corresponding tab 'XML to CMX' needs to be selected. Then the XML file is defined. Note that the folder, in which Commetrix Producer is located needs to contain the Commetrix_Schema.xsd file for the production process to succeed. As described in the previous sections, the parameters are then entered and then, with a click of a button, the network can be generated. Again, for larger networks, this can take up considerable time.

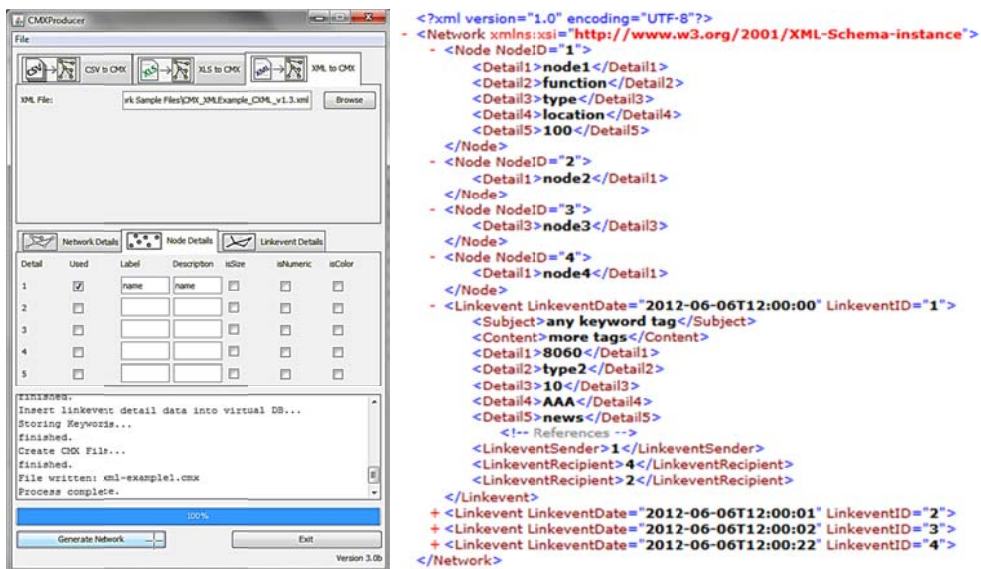


Figure 5: Commetrix .xml Import and an excerpt of an XML example.

Format Requirements. The XML Schema defines the supported XML structure. An XML document can be imported if it validates against this schema. A XML document contains a single network. The root element of the XML tree is the element "Network". A network consists of Node elements and Linkevent

elements. The number of these elements is not restricted. Each Node element must have a NodeID attribute. The value of the attribute must be a positive integer and must be unique in the context of Node elements. Each Linkevent element must have the attributes LinkeventID and LinkeventDate. The value of LinkeventID must be a positive integer and must be unique in the context of Linkevent elements. The value of LinkeventDate must be from the XML Schema type "dateTime" (see <http://www.w3.org/TR/xmlschema-2/#dateTime>). Note: Fractional seconds, timezones, and dates BC are NOT supported. It must be avoided to use dates BC because it causes unintended behavior of the network. The structure of a valid dateTime value is yyyy-mm-ddThh:mm:ss. T is a separator indicating that time-of-day follows. There exist several elements adding properties to nodes and linkevents:

- Node and Linkevent elements may have the subelements Detail1, Detail2, Detail3, Detail4, Detail5. Each detail contains a string value that is restricted to 255 characters.
- A Linkevent element may have the subelements Subject and Content. Each detail contains a string value that is restricted to 255 characters.
- A Linkevent element may have the subelements LinkeventSender, LinkeventRecipient, and LinkeventParent. Each of these elements contains a positive integer value. The values of LinkeventSender and LinkeventRecipient must exist as a NodeID.

The values of LinkeventParent must exist as a LinkeventID.

6 Tables of the Commetrix Data Model (.csv & .xls)

Various tables are used to store network data in the Commetrix data model but not all cells of the tables need to contain data. The special approach is necessary as Commetrix is able to capture much more information than conventional social network models. First of all, Commetrix does not store relationships but *relational events* (which dynamically comprise the relationship). Further, *node attributes* and *link attributes* can be stored (categorical, ranks, metric). An illustrative example is the differentiation of national vs. international links (category variable) or the income of an actor (metric variable). For the link events the user can additionally store including *keywords/tags*, and a *time stamp*. It is hence even possible to use Commetrix to study how different contents related to each other (e.g. relationships in a coding tree, between keywords, between postings, etc.). Also link events can be initiated by one person but targeted at multiple persons. However, also very simple node link networks (without time or properties) can be modeled. The time is then set to an individual time point and not all cells of the tables are filled.

The tables can be created with a spreadsheet tool or, as far as the .csv tables are concerned, even with a simple text editor. The values of a table's row are simply stored as text lines divided by comma (or semicolon, depending on your national convention). Later, in Commetrix Producer, you can select the necessary tables to create a network from 'Comma Separated Value' (.csv) or Excel (.xls) tables. These are standard formats for storing table data. We will now go through the individual required tables (or work sheets in Excel, where you can edit all tables in a single file).

6.1 Node Table

The first table that is part of the Commetrix data model is containing the node data. For the comma separated value table (.csv), a typical name would be node.csv (although you can name it as you like). In the Excel version the data is stored in the worksheet named 'Node'. In the following we will focus on the node.csv, but the Excel worksheet has exactly the same format.

The node.csv (or node worksheet in the Excel version) starts with a header which is always the same. It looks like this (there can be “,” instead of “;”):

nodeID;Detail1;Detail2;Detail3;Detail4;Detail5

There is an example .csv (and .xls) file coming with the Producer. Please do not change or delete the header to have the Producer accept the .csv file (or the .xls file). The header shows the data elements stored for each node. You can use it to interpret the numbers of the following lines, e.g., the first number is the nodeID, the last is Detail5 (explanation follows).

The node.csv (or node worksheet in the Excel version) stores each node of the network in a separate line, for example:

1,1,0,node1,nodetype1,10,,

or, if semicolons are used as separators:

1;1;0;node1;nodetype1;10;;

This is an example text line taken from the node.csv that comes with the distribution version of Commetrix Producer. We can read it as follows: Node with ID number 1 (of network number 1, this is a constant value in the distributed software version) has the name 'node1' (can be shown as label later on). As properties for this node, a nodetype1 is stored (as a category variable, useful for coloring nodes differently), then follows a '10' which could be the importance of that node (i.e. a rank variable or a metric variable). Details 4 and 5 have been left empty (no further properties are stored for the node).

Node ID. The dataset hence starts with a node ID (e.g. 1,2,3,...) to label each node with a number. It must be unique for each node, i.e., ensure that no two nodes have the same NodeID. This number is required.

Detail1 – Detail 5. Commetrix allows you to store up to five node attributes. They can be categories, or rank/metric variables. Categories can be different types of nodes (e.g. 'manager', 'employee', or 'Domain1', 'Domain2'). They can also be non-repeating categories, such as the nodes' names (e.g. 'Name 1', 'Name2',...'NameX'). Ranked categories (e.g. 'low', 'med','high') need to be transformed into ranked numbers (i.e. '1','2','3'). Similarly all sorts of metrics can be stored (e.g. '10','50','78'). In the above example Detail 1 contains a name ('node1'). Detail 2 contains some nodetype ('nodetype1'). Detail 3 contains some metric variable ('10'). Details 4-5 are not in use. Further examples of node properties could be: actor location, actor job position, an evaluation, or a comment.

Empty Elements. Note that empty variables are not visible in the text, but their associated delineators, i.e. the ',' or ';' is noted (see the two subsequent “,” or “;” at the end of the line: this actually means “,(no value),(no value)”).

In short mostly the lines begin with the nodenumber, e.g. 1 - the rest may be left empty, which means only the ',' or ';' appear in a sequence. Make sure that in each line of the node.csv there are five ',' or five ';'.

A minimal example would hence be: “1,,,,,” for node 1 without any attributes.

A more comprehensive example would be: “1,Miller,JM,Manager,90000,male” to store actor number 1 who is named Miller, has the initials JM, is of actor type manager, earns 90000 USD per month, and is male.

6.2 Linkevent Table

Commetrix has the linkevent as its most basic unit of interrelationship between nodes (not the relationship link as many other SNA tools). Linkevents are the different events (or just one) that constitute the relationship. E.g. two people may have one link based on their interaction, which is consisting of 10 exchanged mails. Here we have 10 link events, i.e. the mails (yielding a relationship of strength 10).

You hence need to consider how to model your network data as a set of network events. But this is actually really simple. For example, if you just have a set of nodes linked with binary relationships (absent or present), then you create the nodes in the nodes.csv table and the linkevents are just fictive events that establish the link at a certain fictive time. Note however that you need to define an initiator (and optionally) a recipient of the link. There are measures in the Analyzer, later on, which ignore the directedness. Alternatively you can also model a link event from A to B and one from B to A to create a reciprocal link between the two nodes. There can be linkevents which have no recipient (e.g. an initiated message by some actor). However, they will not appear in the network visualization or analysis.

The linkevent.csv (or linkevent worksheet in the Excel version) starts with a header which is always the same. It looks like this (there can be “,” instead of “;”):

linkeventID,linkeventDate,Subject,Content,Detail1,Detail2,Detail3,Detail4,Detail5

Example. The linkevent.csv (or node worksheet in the Excel version) stores each node of the network in a separate line, like for example:

1,2012-05-06 12:00:00,document submission,Here is my doc,project,international,3,3,0

This line documents a linkevent number 1 sent on May,06,2012 at 12:00 noon, with the heading (optional) “document submission”, the content “Here is my doc”, and finally five linkevent properties, e.g., the domain “project” (later interpreted as a category variable useful for filtering out project related interaction), then, the scope of the exchange “international” (category variable), also coded as “3” (to make it a rank variable with 1 = local), and finally a sentiment coding of “0” for neutral message.

Note that not all fields need to be filled out in every case. The minimum requirement for storing linkevents is to store a unique linkevent ID number and a date in the prescribed format (YYYY-MM-DD hh:mm:ss, e.g. ‘2012-06-06 12:00:00’ without the quotes) as in: “1,2012-06-06 12:00:00,,,,,”. If no time information is available or there is just one snapshot of the network, simply use some arbitrary and identical timestamp for all linkevents (e.g. today). This will create the network at once, without any change or evolution. You could also consider using fictive time stamps and increment them (e.g. the next link event is 1 day after the previous. This will allow you to scroll through your network (adding links individually in a sequence).

Linkevent ID. Each linkevent needs to be assigned with a node ID (e.g. 1,2,3,...) to label each node with a number. It must be unique for each linkevent, i.e., ensure that no two linkevents have the same LinkeventID. This number is required.

LinkeventDate. Each linkevent needs to have a timestamp (i.e. a date) in the prescribed format YYYY-MM-DD hh:mm:ss, e.g. ‘2012-06-06 12:00:00’ without the quotes. If you are not interested in temporal aspects, just take one selected date (e.g. today) for all your linkevents. The network is then created at one single point in time and remains constant. Sometimes, however, it might be useful to introduce an artificial temporal sequence (e.g. each new linkevent is added an hour later). This will later allow you to

play through the unfolding network (you still have the option to look at the total modeled network, of course).

Two link events can be happening at the same time. If you want to create a strong vs. a weak relationship, you could just create many linkevents at a certain time between two nodes. Alternatively you can store the link strength in a variable (Detail 1-5) and later filter out all linkevents with the desired strengths (e.g. using the keyword filter, explained in the Commetrix Analyzer documentation). If your event connects many nodes (e.g. three actors sharing a document to be modeled as a completely connected triangle), then model 1 linkevent from actor A to B, one from B to C, and one from A to C, each with new linkevent IDs.

Note that in Commetrix, linkevents do not naturally disappear, i.e., the network constantly grows. Also you cannot model the termination of a relationship (i.e. a negative event). But you can model constant activity (e.g. mails) as reoccurring linkevents in order to signal that a relationship is constantly (re-) activated/growing and use the sliding window mode of Commetrix Analyzer, in which links are blended out if their underlying linkevents are outside a defined time period (e.g. you could define that a collaboration link has to be ‘true’ for each month by assigning a new linkevent for each month indicating this fact; in a 3 month time window mode the collaboration link is not going to be shown if no collaboration (re-activation) was coded for the last 3 month).

Subject and Content. Commetrix offers a blend of topic analysis and network analysis. You can store keyword descriptors or all sorts of variable terms in the two text fields subject and content. They are later concatenated in the analysis so it suffices if you fill out only one of them. You can store for example e-mail headers and textbodies, wiki-article titles and articles, observations, tags, etc. The fields can also be left blank, if you are not having data about the contents of a network interaction.

The fields can also be used to store category tags for the captured interaction events, e.g. “project”, “on time”, “friendly”, “international”, “phone interaction”, etc. (without the quotes). It might be useful to mark such ‘artificially’ assigned tags with some symbol, e.g. "!international", or "!online". Note the ‘!’ in the beginning to mark that it is no real word but meant as a category. This is especially necessary if you have both, real contents (e.g. texts) and your assigned codes. Later in the keyword list (in Commetrix Analyzer) you can easily spot your codes, also, if you use ‘!’ to mark them the codes will appear at the top of the list. You can later filter the network to include only those links which contain one (or more) such descriptors. They can however not be used for coloring or sizing (they are not assumed to form categories of nominal or metric types).

Detail 1-5. Commetrix allows you to store up to five node attributes. They can be categories, or rank/metric variables. Categories can be different types of linkevents (e.g. ‘friendly’, ‘hostile’, or ‘Domain1’, ‘Domain2’). They can also be non-repeating categories, such as the linkevent names (e.g. ‘Name 1’, ‘Name2’,...‘NameX’). Ranked categories (e.g. ‘low’, ‘med’, ‘high’) need to be transformed into ranked numbers (i.e. ‘1’, ‘2’, ‘3’). Similarly all sorts of metrics can be stored (e.g. ‘10’, ‘50’, ‘78’). In the above example Detail 1 contains a category (‘project’). Detail 2 contains some linkeventtype (‘international’). Detail 3 contains a rank form of the internationality categories (‘3’), etc.

Note that the sender and recipient is not modeled in this table. This is due to the fact that a link event can have multiple senders or recipients (and also none). In Commetrix the actual assignment of sending and receiving nodes is done separately in the following tables.

6.3 Linkeventsender Table

The sender/initiator and also the recipient(s) of a linkevent are stored in the node.csv table (or in the Excel worksheet). If you have not done this yet, create the according node in the table first and assign an

ID, as described above. The node IDs are needed now to assign nodes as senders or recipients of link events (referenced by their linkevent ID), in the linkeventsender and linkeventrecipient tables.

Example. Let us look at an excerpt of an linkeventsender table. The first line contains the standard header. It may not be changed. The second and third lines are an example of how nodes are assigned as initiators of linkevents.

```
linkeventID;senderNodeID 1;1  
2;9
```

Generally, initiating (e.g. sending) a linkevent is captured by documenting the linkeventID, noting the ID of the initiating node and in the third line, for example, linkevent number 2 (which could be a message that we stored in the linkevent table) was initiated by node number 9 (which could be some actor stored in the node table). The third number is denoting the network number (we just have one, so it is '1').

Note that each linkevent and node ID needs to be consistent with your entries in the linkevent table: they have to exist there as well.

6.4 Linkeventrecipient Table

The linkeventrecipient is coded in exactly the same way as the sender. In our example, the first line contains the standard header. It may not be changed. The second and third lines are an example of how nodes are assigned as initiators of linkevents:

```
linkeventID;recipientNodeID;  
1;5  
1;4
```

Note that instead of the initiator of a particular linkevent, we now need to code the recipient. Multiple recipients can be modeled as well. In the example, linkevent number 1 was received by node number 5 and 4. Again, note that each linkevent and node ID needs to be consistent with your entries in the linkevent table: they have to exist there as well.

6.5 Linkeventparent Table

Interaction can also follow a different mechanism, where one relational event (e.g. a message) is referencing another relational event (e.g. a previous message). A very common example is a discussion, e.g. an online discussion or documents referencing other documents. In such a domain it is often difficult to identify recipients (i.e. looking 'forward' from the initiators perspective). In such a situation, Commetrix offers to capture, which linkevent referenced a PREVIOUS other linkevent. For example, in a document I reference other existing documents which served as my foundation, or, in a discussion I reference a previous posting (i.e. the initiator looks 'backwards' in time). Of course, each referenced linkevent should have an author assigned, e.g. the author of the discussion post that I am referring to. In this way, indirectly, a link is established between two nodes, e.g., message 2 of author 2 is referencing a previous parent message 1 of author 1, in other words, author 2 is referencing author 1. This scenario is captured in the linkeventparent table, in which one linkevent is related to another (usually preceding) linkevent.

Here is an example with the standard header in the first line and then two lines of data:

```

linkeventID:parentLinkeventID
5:4
6:5

```

If we take line two of this excerpt, the linkevent (e.g. a message) with ID number 5 is referencing a parent linkevent with ID number 4. These linkevents will only be visible in the network if a node is assigned as their initiator. Assume node 2 is assigned to linkeventID 5 and node 1 is assigned to linkeventID 4. In the network a line would be drawn from node 5 to 4, based on the linkevent reference modeled in the above table. This also implies that the linkeventparent table sort of replaces the linkeventrecipient table, but a linkeventsender table is necessary to be filled with data.

The sender/initiator and also the recipient(s) of a linkevent are stored in the node.csv table (or in the Excel worksheet). If you have not done this yet, create the according node in the table first and assign an ID, as described above. The node IDs are needed now to assign nodes as senders or recipients of link events (referenced by their linkevent ID), in the linkeventsender and linkeventrecipient tables.

7 Defining Import Parameters

Before the network is actually generated (see below), a series of parameters is to be set. This is done in the middle section of the user interface of Commetrix Producer. There you find three tabs: (1) Network Details, (2) Node Details, (3) Linkevent Details.

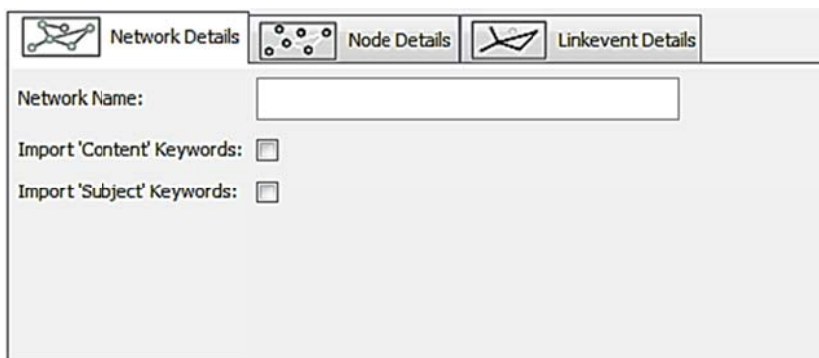


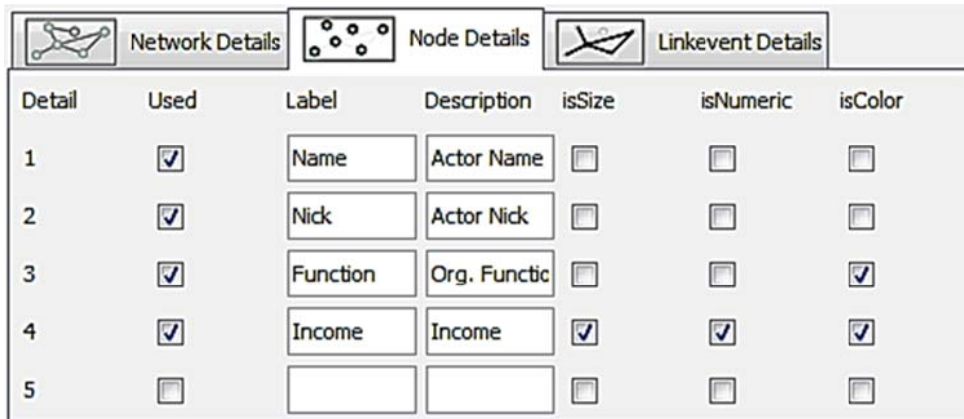
Figure 6: Defining Network Details – User Interface.

7.1 Network Parameters

Define Network Name. A first step is to define a name for the .cmx network file that is to be produced. It needs to be entered into the ‘Network Name:’ text box.

Import Keywords. The user might want to employ a text mining process. A simple text mining process is available via button ‘Import Keywords’. This process splits the entries in columns ‘Subject’ and ‘Content’ of table ‘Linkevent’ into single tokens (keywords) and removes stopwords as well as too short keywords from the data set. The corresponding checkboxes need to be checked, if the data stored in the Content and Subject columns of the Linkevent table should be considered. The process results in a list of keywords that can later, in Commetrix Analyzer, be selected as filters (e.g. project names, tags, etc.). However, also consider that this feature can take much memory and resources so it is sometimes advisable to not import keywords when working with larger datasets.

7.2 Node Parameters

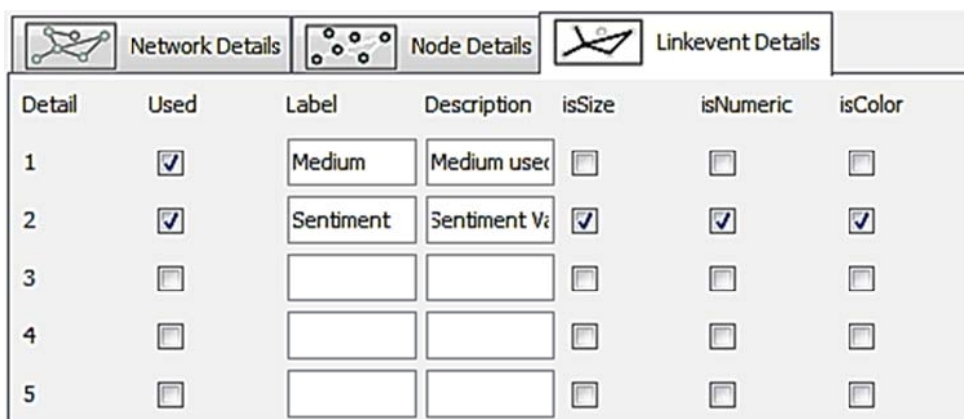


| Detail | Used | Label | Description | isSize | isNumeric | isColor |
|--------|-------------------------------------|----------|---------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1 | <input checked="" type="checkbox"/> | Name | Actor Name | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 | <input checked="" type="checkbox"/> | Nick | Actor Nick | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 3 | <input checked="" type="checkbox"/> | Function | Org. Function | <input type="checkbox"/> | <input type="checkbox"/> | <input checked="" type="checkbox"/> |
| 4 | <input checked="" type="checkbox"/> | Income | Income | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 5 | <input type="checkbox"/> | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Figure 7: Defining Node Details – User Interface.

Node Details. In a next step the Producer needs to know the definition and variable types of the node details that you have coded. The Figure shows an example. The user marked that Detail 1 is ‘Used’. Commetrix Analyzer will create the Label ‘Name’, e.g. to give the analyst the option to choose ‘Name’ as a label in the network graph. A description is also helpful and shown in the Analyzer. Next the variable properties need to be stated. Generally, all variables can be used as labels (automatically, if defined here). Further we can denote, whether the coded property/detail can be interpreted as a node size (check ‘isSize’, whether they are numeric (e.g. required for gradient coloring from blue to orange), or whether the node property shall be assigned a color (mark ‘isColor’).

In the above example, detail1 can only be used as a label (it is a ‘non-repetitive category system’). However, sometimes it might be desirable to give each node name a color. In this case ‘isColor’ should also be marked. The contents of the variable stored in detail1 are, however, not numeric and cannot be interpreted meaningfully as a node size. Therefore we find no corresponding marks. In contrast, the income, modelled as variable ‘detail4’ in the node.csv (or Excel worksheet), are used as a label next to the node in the graph, interpreted as a node size (higher income yields larger nodes), is defined as numeric (i.e. metric) which yields, in combination with marking ‘isColor’ the opportunity to use a color gradient which results in blue nodes for low income and orange nodes for high income (and a gradient between the extreme values). The best is to experiment a bit with these features.



| Detail | Used | Label | Description | isSize | isNumeric | isColor |
|--------|-------------------------------------|-----------|--------------|-------------------------------------|-------------------------------------|-------------------------------------|
| 1 | <input checked="" type="checkbox"/> | Medium | Medium user | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 2 | <input checked="" type="checkbox"/> | Sentiment | Sentiment Va | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| 3 | <input type="checkbox"/> | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 4 | <input type="checkbox"/> | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| 5 | <input type="checkbox"/> | | | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Figure 8: Defining Node Details – User Interface.

Linkevent Details. Linkevent Details are defined in a similar way as node details. We can define which of the stored variables (in the detail1-5 columns) are to be used in the production process. We assign a label to the variable and a description. Finally, we need to classify the properties of the variables. Is it only to be used as a label (check no box), or is it to be interpreted as a color (check 'isColor'), as a color gradient (check 'isColor' and 'isNumeric') or as a link width (check 'isSize').

8 Starting the Production Process

After the Import Documents (.csv and .xls) have been prepared and are selected in the corresponding text boxes of the Commetrix Producer (sections 5 and 6) and after the Import Parameters have been set (section 7), the production process can finally be launched. This is simply done by pressing 'Generate Network'. Note that depending on the size of the dataset this process can take quite long.

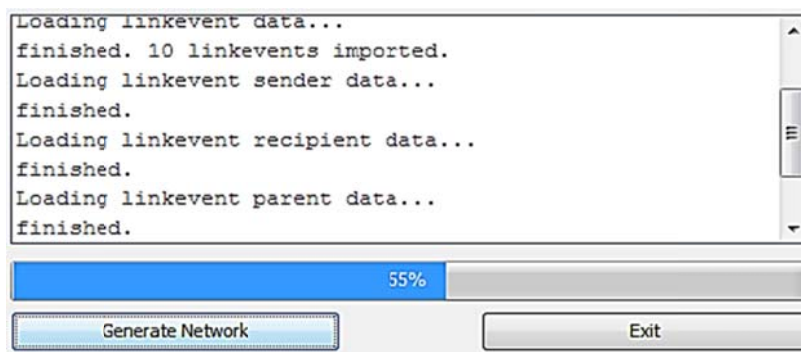


Figure 9: Launching the network production process.

After the process is finished, a .cmx file with the name defined in the Network Details tab is generated and stored in the directory in which the Commetrix Producer was installed. This file can now be used for analysis with Commetrix Analyzer. Please consult the Analyzer manual for more information about the available features, such as content filtering, temporal evolution, ego-network detection, 3D views, network analysis (over time) and much more.

9 Contact

If our Commetrix Framework Solution is of interest for you, we will be happy to arrange a license contract. Just contact us via: info@trilexis.com (preferred) or via the following address.



Trilexis Information Management GmbH

Helmut-Kaeutner-Weg 5,

13591 Berlin

GERMANY

Corp. Registration ID: DE 29 039 00437

info@trilexis.com