JDisc Discovery 5.0

Dependency Mapping Add-On

Legal Notice

Copyright

JDisc GmbH
Kuppinger Weg 25
D-71116 Gärtringen
Germany

Contents

1 Introduction

The introduction chapter explains what dependency mapping is all about and how JDisc Discovery discovers dependencies between devices on the network.

1.1 What Is Dependency Mapping?

Dependency mapping is about finding connections between devices on the network. Devices communicate with each and frequently also depend on each other. The diagram below illustrates a web-based application consisting of a database server, a web server, an application server and a user interface client.

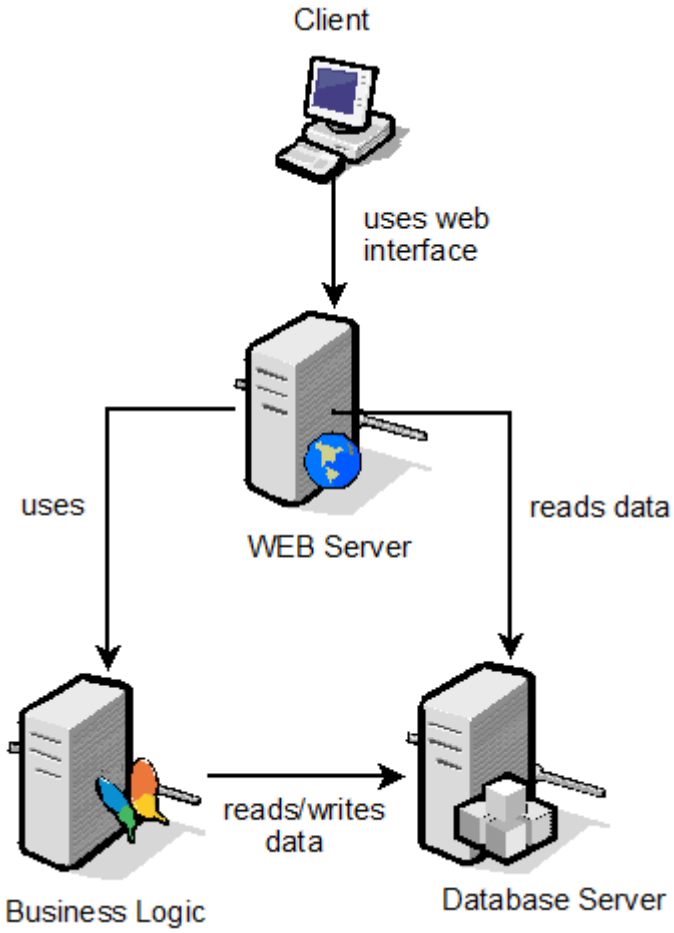

Fig: A typical 3-tier web application

- The client uses the web interface to access the web server.
- The web server accesses a database on the database server and the business logic on the application server.
- The application server (hosting the business logic) reads and writes to the database

on the database server. It *depends* on the database server.

Knowing dependencies between devices is crucial for many IT projects.

For example, suppose your company decides to virtualize servers in the data center to reduce IT expenses. Instead of having three physical servers you would finally run only one physical server hosting three virtual machines.

In such a case it is important to know the dependencies between servers to sequence the migration of the servers in the right order. If you would migrate a server that is still needed by other servers, business applications might fail.

2 Installation

Installing the JDisc Discovery dependency mapping add-on requires JDisc Discovery 5.0. To install the dependency mapping add-on, double click the MSI file. The installation wizard will guide you through the installation process.

The dependency mapping add-on installation restarts the JDisc Discovery service.

3 Dependency Mapping

This chapter explains how to get started with the dependency mapping add-on.

3.1 Licensing

The JDisc Discovery dependency mapping requires its own license option. From the JDisc Discovery user interface, select *Help* » *License Info* to display license information. The Dependency Mapping add-on must appear in the *Options* table.

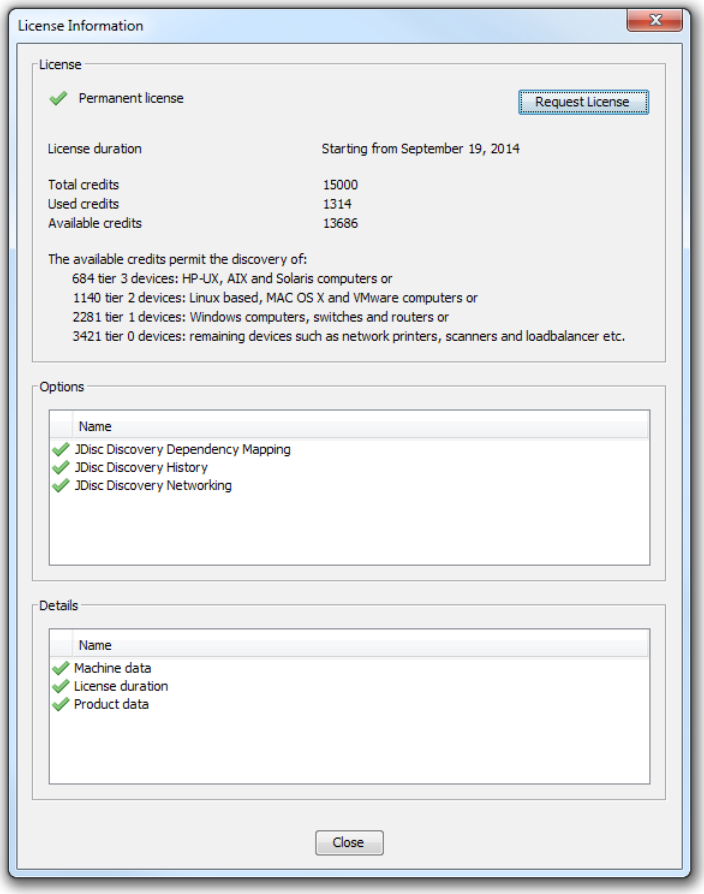

Fig: License Information dialog

3.2 How JDisc Discovery Collects Dependency Information

JDisc Discovery collects connection information from many different device types. Devices supporting SNMP typically provide connection information via the *TcpConnTable* and *UdpConnTable* for TCP and UDP connections.

Often computers do not support SNMP. In this case, JDisc Discovery collects connection information using JDisc Discovery's remote login capability. All major operating systems implement the *netstat* command. The *netstat* command displays active TCP/IP connections as well as open TCP and UDP ports.

To collect dependency information, remote login must be enabled for devices that do not support SNMP. Refer to JDisc Discovery's user manual for more information on how to enable remote login.

Please note: When JDisc Discovery discovers a network, the device connections only represents a snapshot of the network. However, when you configure JDisc Discovery to run scheduled discovery jobs, over time the dependencies between devices on your network will become more accurate.

JDisc Discovery does not discover connections originating from the computer that runs the software. This is because the discovery running on this computer opens many network connections to other devices on the network.

3.3 New User Interface Elements

The installation program adds new elements to JDisc Discovery's user interface:

- A new menu item *Port Names...* in the *Discovery* menu.
- A new menu *Connections* in the *Networking* menu.
- A new top-level menu *Maps.*
- A new menu item *Create Dependency Map...* in the device report's context menu.
- A new tab *Connections* in the device details dialog.
- A new *Connections* panel in the discovery settings dialog's *Data Collection* tab.

3.4 Configuration

Open the discovery configuration dialog from *Discovery* » *Configuration* menu and select the *Data Collection* tab. The dependency discovery is enabled by default.

Per default, connected devices will not be discovered but only created with IP addresses in the database. Enable the *Discover connected devices* option to also discover connected devices. Connected devices might be outside of the discovery scope (networks, address ranges, Windows domains or directory objects).

Connected devices must not necessarily reside in your corporate network!
Use JDisc Discovery's IP address filtering to restrict the discovery scope to addresses within your corporate network. Refer to the JDisc Discovery user manual for more information on IP address filtering.

If existing connections (stored in the database) do not show up again in subsequent discoveries for a defined period of time these can be deleted automatically. Select the *Delete inactive dependencies between computers when older than XXX day(s)* option and enter the desired number of days. If this option is disabled, connections are never aged out.

Of times computers expose the date and time since the last reboot. JDisc Discovery uses this last reboot date to delete all connections that have been established prior to the last reboot. Those connections are definitely closed, since a reboot closes all network connections.

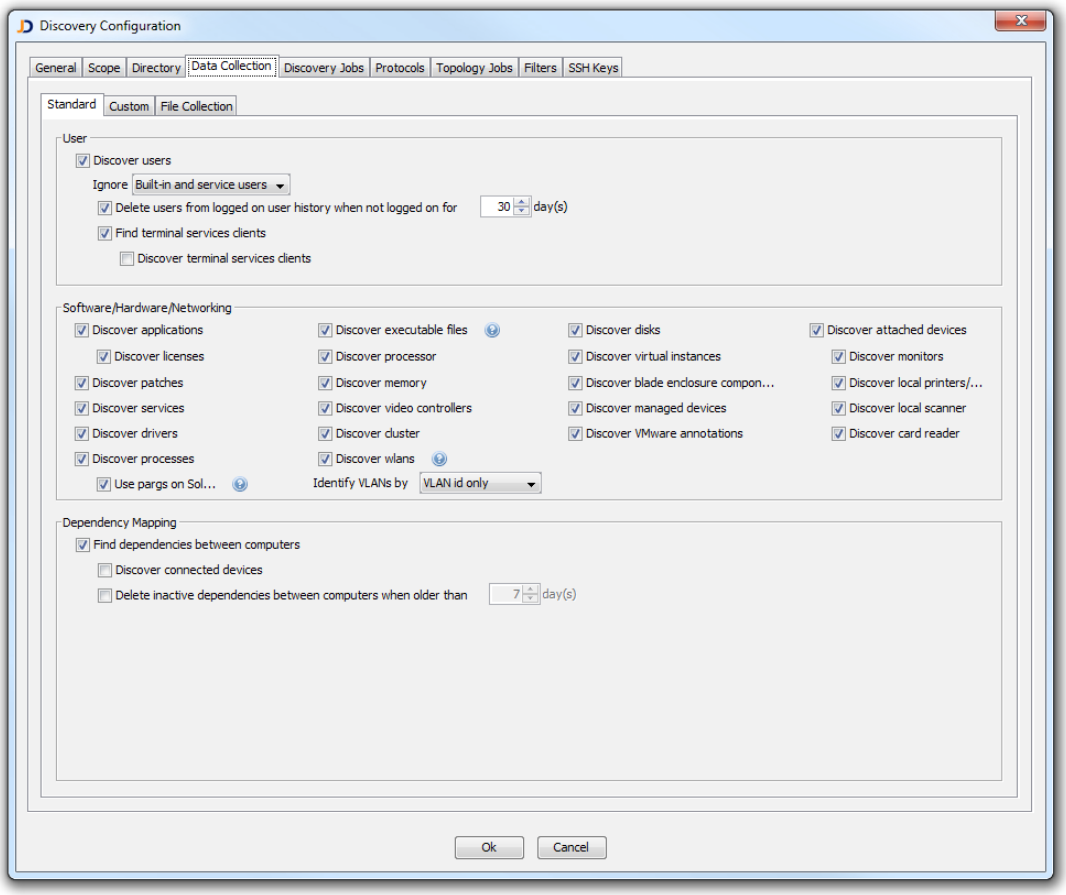

Fig: Dependency mapping configuration

3.5 Define Port Names

JDisc Discovery assigns symbolic names to port numbers based on well known ports. If your corporation uses different port assignment or if port assignments are missing, open the *Port Names* dialog from the *Discovery » Port Names* menu item.

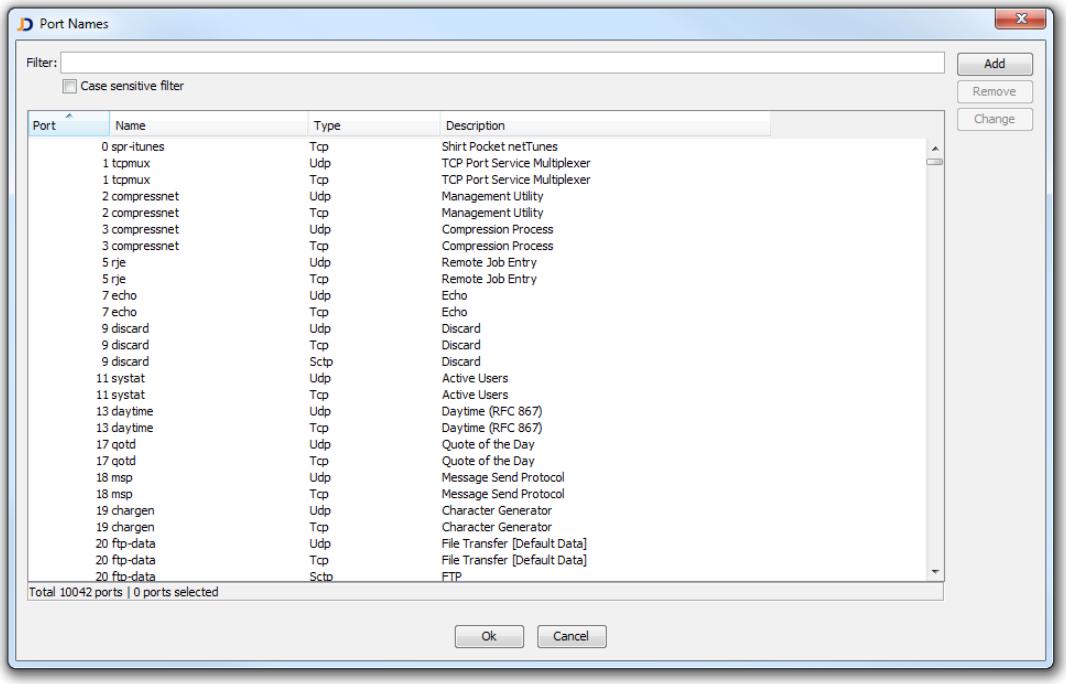

Fig: Port name configuration

Select a port, double click or click *Change* to modify existing port assignments. Click *Add* to add new port assignments or *Remove* to delete port assignments.

3.6 New Reports

The installation program adds new reports to JDisc Discovery's reporting system.

3.6.1 Open Port Statistics

The *Open Ports* report displays all open ports including port name, port type and device count. Open the Open Ports report from *Networking » Connections » Open Ports*.

From the *Open Ports* report, select one or more ports, open the context menu and choose one of the menu items below:

- *Devices listening on selected port(s)*
- Devices connecting to selected port(s)

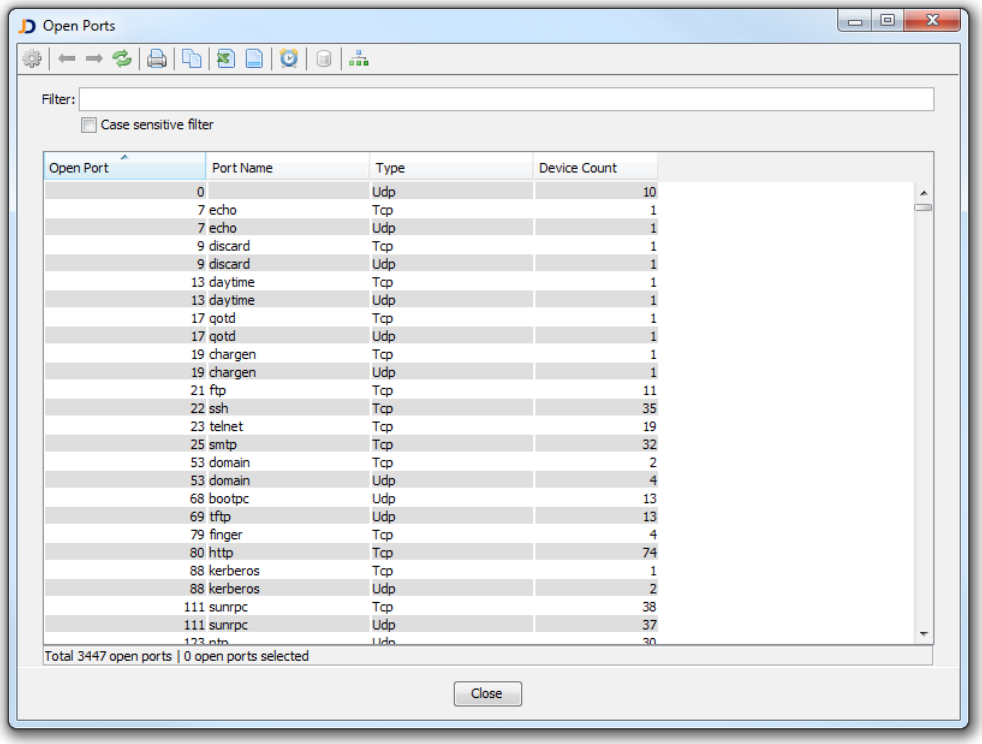

Fig: Open port statistics

3.6.2 Connection Statistics

The *Connection Statistics* report displays all devices including the number of open ports and connections. Open the *Connection Statistics* report from *Networking » Connections » Connection Statistics*.

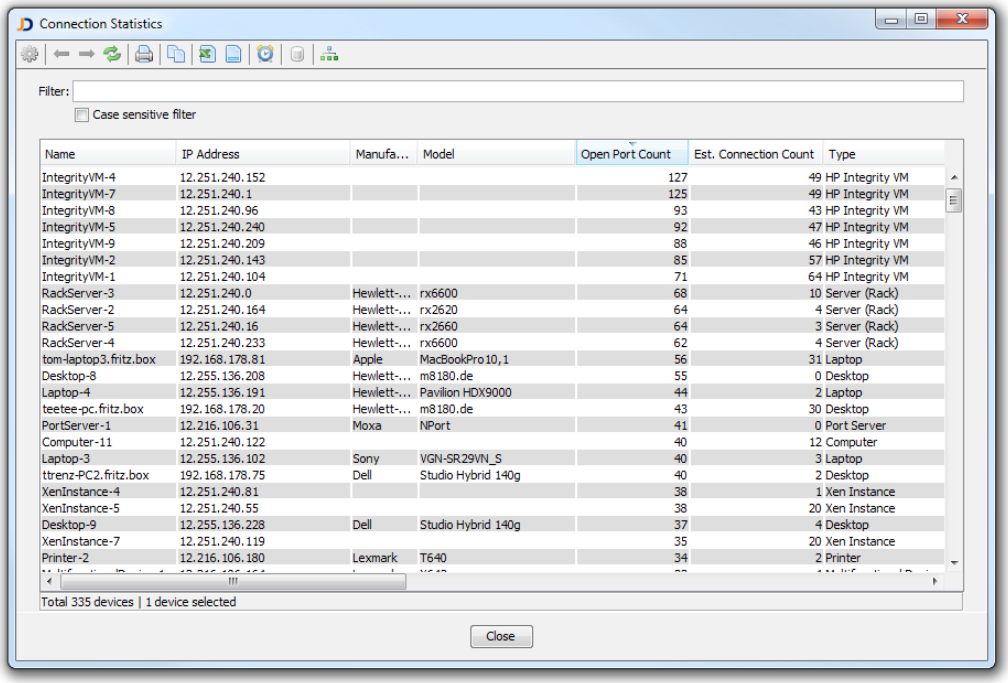

Fig: Connection statistics

3.6.3 Device Details - Open Ports And Connections

JDisc Discovery extends the *Device Details* report with a new tab called *Connections*. The *Connections* tab contains the *Open Ports* and *Connections* reports.

The *Open Ports* report displays all listening ports of the selected device.

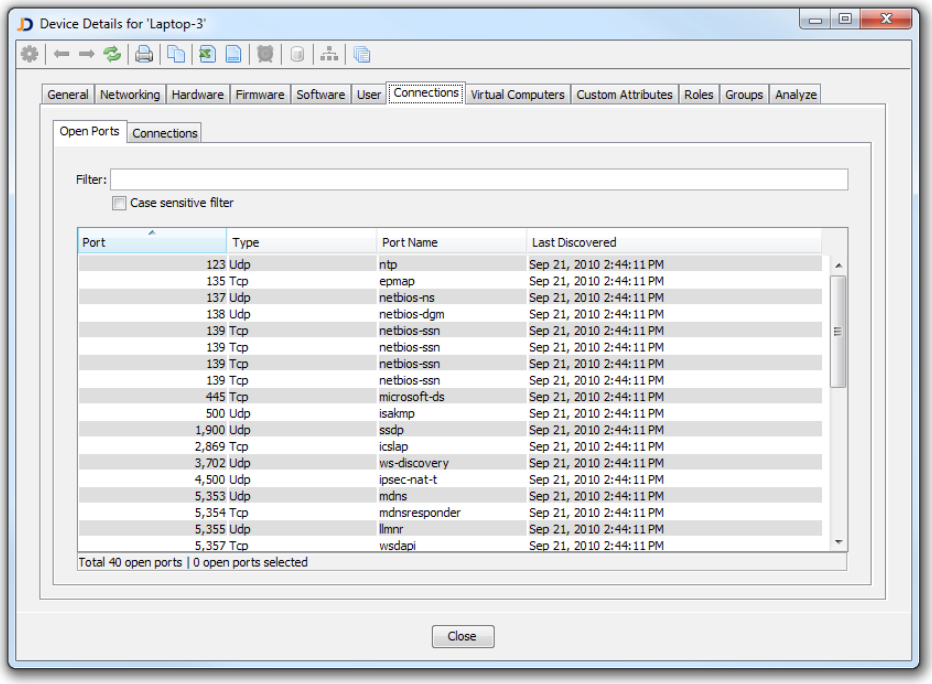

Fig: Open ports

The *Connections* report displays all TCP/IP connections of the selected device.

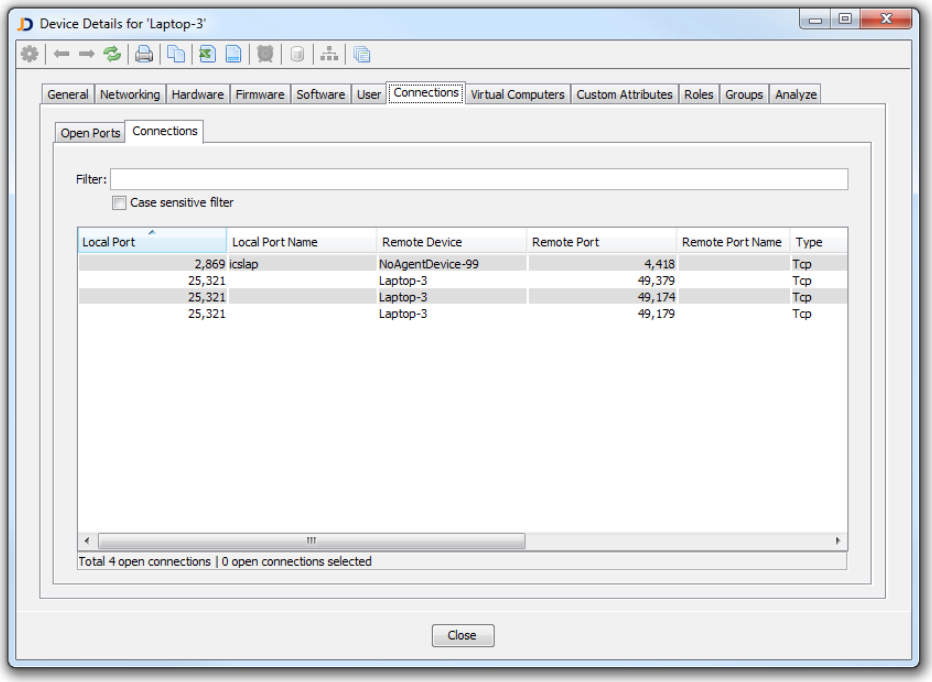

Fig: Device connections

3.7 Dependency Maps

Dependency maps graphically represents connections between devices. A dependency map displays the devices and the connections to other devices.

3.7.1 Dependency Map Basics

A dependency maps must always consist of one or multiple root devices. Root devices are the starting point for displaying dependencies to other devices. Root devices are displayed blue colored in dependency maps.

Dependency maps can contain many devices in large corporate networks. To improve overview, reduce loading time and to adapt dependency maps to your specific use cases, you can filter devices by:

- Connection hop limit - Only displays connections and devices that connect to any of the root devices with not more than the specified hop count.
- Port filter - Only displays connections and devices that connect to each other with selected ports.

- Type filter - Only displays devices of selected device types.

In addition to the above filters, devices can also be manually removed from a dependency map.

Frequently used dependency maps can be saved and opened from the *Maps* menu. JDisc Discovery does not only save the root devices and the filter configuration, but also the location of all devices within the dependency map. You can move devices within the dependency map to adapt the layout to your needs.

3.7.2 Create A Dependency Map

You can create a dependency map from any device report. Select one or more devices, open the context menu and select *Create Dependency Map.* JDisc Discovery opens a new Dependency Map window. Root devices are indicated by blue circles. The connection hop limit is set to 1 by default to prevent the dependency map from become too crowded. Because of the initial (default) connection hop limit, new dependency maps only display directly connected devices.

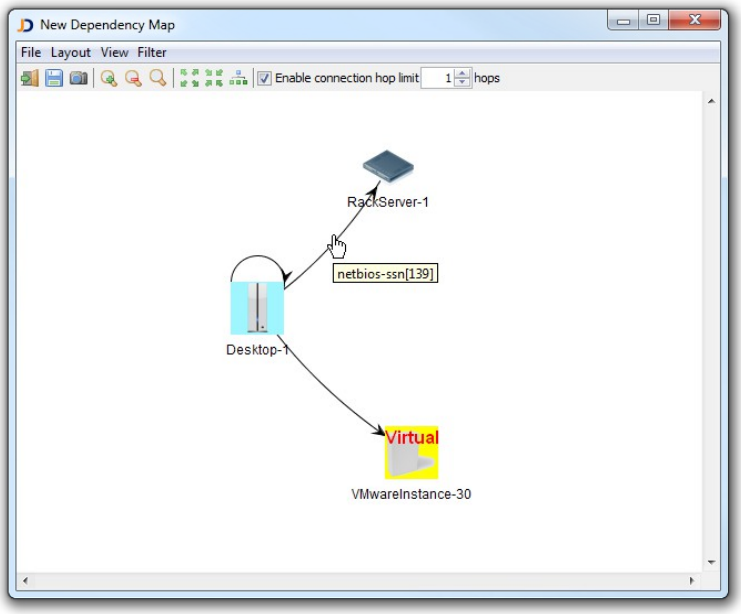

Fig: Dependency map

A dependency map consists of devices (displayed as circles) and connections (displayed as lines) connecting devices. Each line can represent one or multiple connections. Tool tips provide additional connection information. Double clicking a connection opens a new dialog displaying all connections in detail.

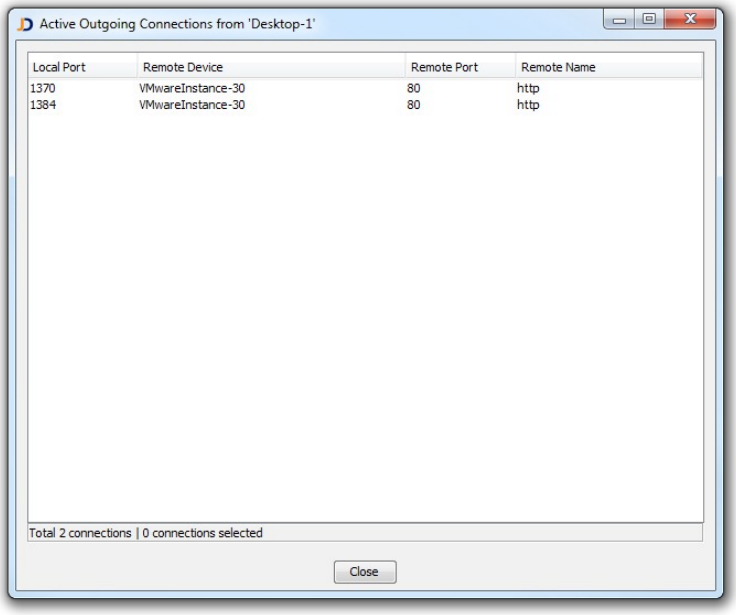

Fig: Active outgoing connections

You can edit port names in the dialog. However this will not change the global port assignments!

Port names can be modified for individual connections without changing the default port assignments.

3.7.3 Change Map Settings

This section explains how to modify and adjust the dependency map configuration to accommodate your needs.

3.7.3.1 Change The Connection Hop Limit

The connection hop limit defines the maximum distance (hop count) from any device on the map to any of the root devices. The default connection limit is set to one - thus only directly connected devices appear on the dependency map. You can change the connection hop limit by directly entering the desired hop count or by using the spinner buttons to increment or decrement the hop count from within the tool bar.

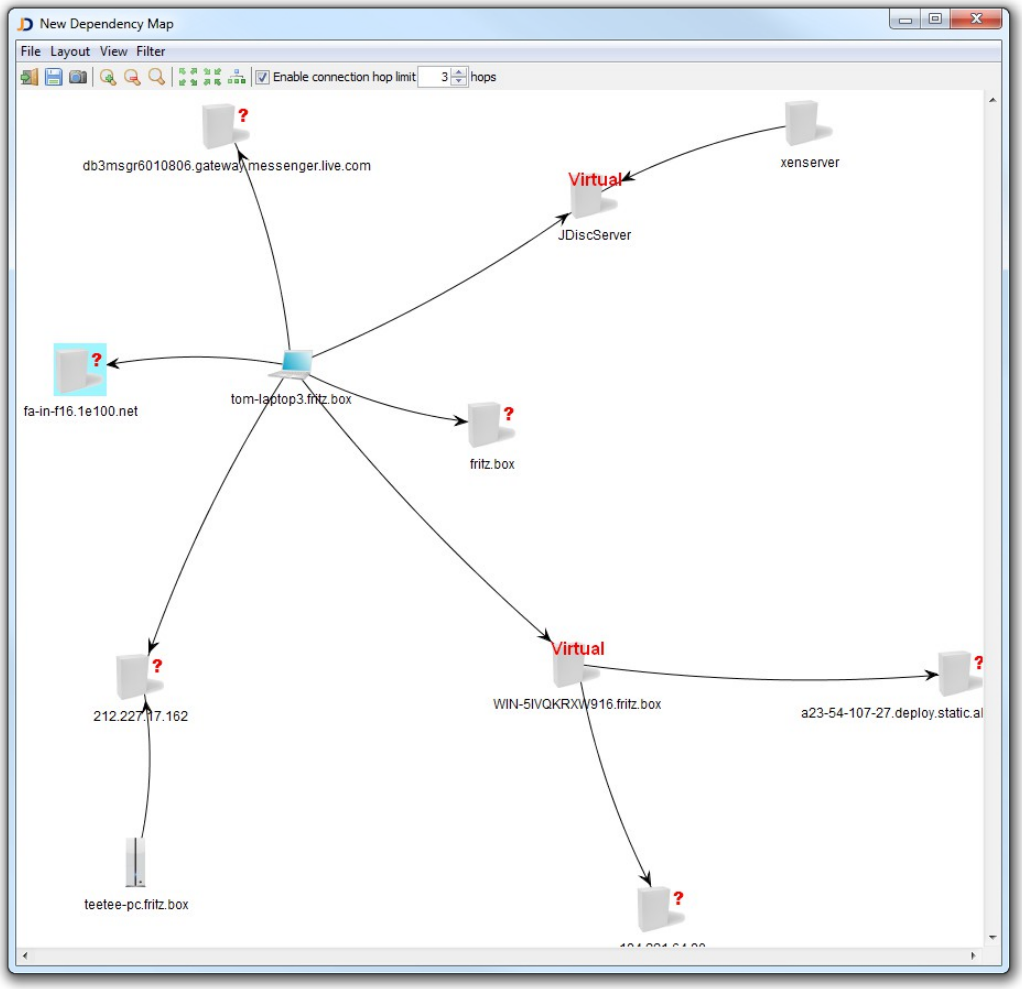

Fig: Change the hop count limit

Changing the hop count limit changes the layout of your map!

3.7.3.2 Filter Dependency Maps

Dependency maps can become very crowded in corporate networks. This is why JDisc Discovery provides configurable port and device type filters to display only connections and devices of interest.

Port Filters

Port filters display only devices connecting to each other with configured ports. To create a port filter, open the *Ports Filters* dialog from the dependency map using the *Filter » Port Filter...* menu item.

The *Port Filters* dialog displays only port assignments for existing connections of devices within the dependency map.

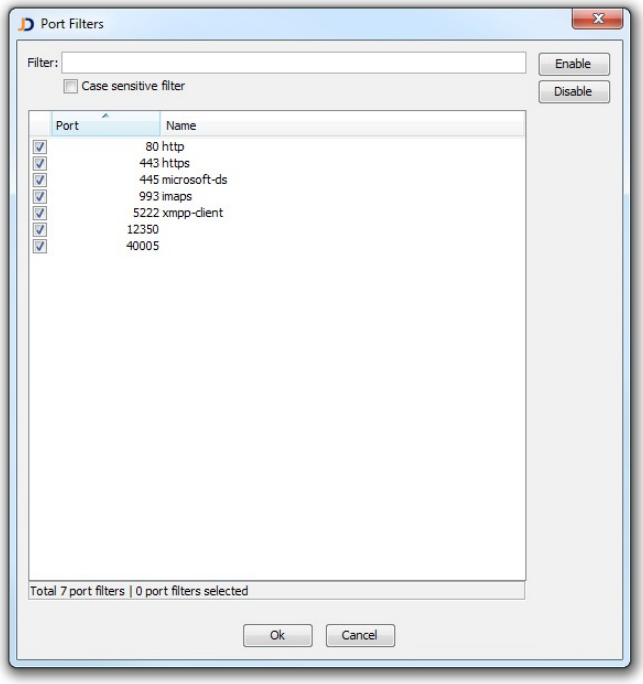

Fig: Port filters

Enable or disable port assignments as needed and click *Ok* to refresh the dependency map.

Changing port filters will re-layout the dependency map.

Device Type Filters

Device type filter only display devices that match the configured device types. To create a device type filter, open the *Device Type Filters* dialog from the dependency map using the *Filter » Device Type Filter...* menu item.

The *Device Type Filters* dialog displays JDisc Discovery's device type taxonomy.

Enable or disable the device types of interest.

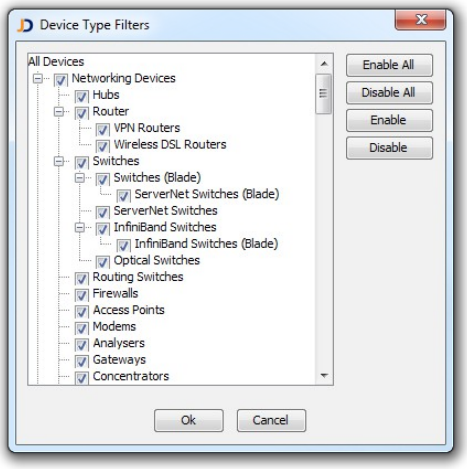

Fig: Device type filters

Disabling a super-ordinate device types does not disable sub-ordinate device types!

Changing device type filters will re-layout the dependency map.

3.7.3.3 Layout Algorithms

JDisc Discovery offers the layout algorithms below for use with dependency maps:

- Kamada-Kawai
- Fruchtermann-Rheingold
- Self-organizing map
- Spring layout
- Circular layout

From the dependency map windows, you can change the layout from the *Layout* menu.

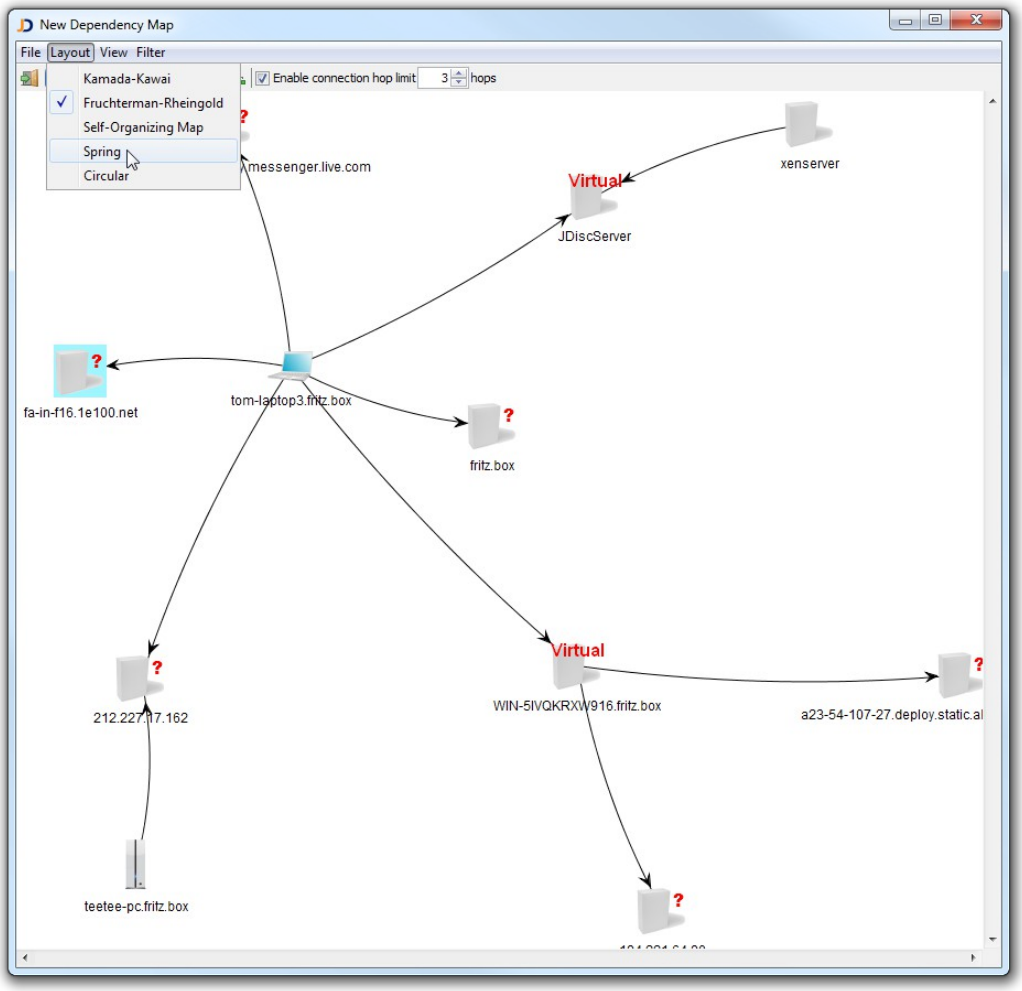

Fig: The layout menu

3.7.3.4 Move Devices

To improve the layout, you can move devices on the dependency map. JDisc Discovery keeps the new layout positions when saving the map.

3.7.3.5 Enlarge / Shrink The Canvas

The canvas might become too small or large if too many or few devices are displayed on the dependency map. You can use the enlarge canvas or shrink canvas icons from the tool bar to enlarge or shrink the canvas size. You can also the *Enlarge Canvas* and *Shrink Canvas* menu items from the *View* menu.

4 Open Source

This software includes software developed by various open-source projects and organizations as listed below. The corresponding files and components are copyright to the corresponding organization or vendor and all rights reserved. The software files and components distributed under the open-source licenses are distributed on an "AS IS" basis, WITHOUT WARRANTY OF ANY KIND, either express or implied. See the license of the corresponding project for specific rights and limitations under the license. Depending on the license, any product derived from the products may not be called with the name of the project nor may the name of the project appear in their name, without prior written permission. For written permission, please contact the corresponding project owner by visiting the corresponding project home page as listed below.

All license files can be found in the installation directory 'Licenses'.

- This product includes software developed by the Apache Foundation (http://www.apache.org). These are 'Axis', 'Commons Collections', 'Commons Net', 'CXF', 'log4j', and 'POI', 'Drools', 'log4j'.
- This product includes the 'SBLIM' WBEM implementation (http://sourceforge.net/projects/sblim/files/sblim-cim-client2/)
- This product includes icons from 'FAMFAMFAM' icon gallery 'SILK' (http://www.famfamfam.com/lab/icons/silk).
- This product includes Kai Toedter's 'Jcalendar' (http://www.toedter.com/en/jcalendar/index.html).
- This product includes the JUNG layout library (http://jung.sourceforge.net).
- This product includes the COLT numeric library (http://acs.lbl.gov/~hoschek/colt).
- This product uses the Postgres database (http://www.postgresql.org).
- This product uses SNMP4J (http://www.snmp4j.org).
- This product uses the Ganymed SSH library (http://www.ganymed.ethz.ch/ssh2).
- This product uses the drools rule engine (http://jboss.org/drools).
- This product uses the janino compiler (http://www.janino.net).
- This product uses Jyhton (http://www.jython.org/Project).
- The product calls the dmidecode binary (http://www.nongnu.org/dmidecode).
  Find the source code in the 'sources' directory.
- This product uses icons from 'Crystal Clear' (http://commons.wikimedia.org/wiki/Crystal_Clear).
- This product uses the 'PUTTY' ssh client.
- This product uses the dom4j library (http://dom4j.sourceforge.net/dom4j-1.6.1).
- This product uses the Jaxen library (http://jaxen.org/)

- This product uses the Jcalendar library (http://toedter.com/jcalendar/).
- This product uses the Jdom library (http://www.jdom.org/).
- This product uses the saxpath library (http://www.saxpath.org/).
- This product uses the miglayout library (http://www.miglayout.com/).
- This product uses the taskdialog library (https://code.google.com/p/oxbow/).
- This product uses the vijava library (http://vijava.sourceforge.net/).
- This product uses the dnsjava library (http://www.dnsjava.org/).
- This product uses the trove library (http://trove.starlight-systems.com/).