Connect Installation Guide

LABVANTAGE Connect version 8.0.0

Document version 2.1

Connect

Solution for instrument and system interfaces

by Software Point Oy

Connect is a solution for interfacing instruments and systems with LIMS. Instrument and system interfaces developed with Connect can be reused and thus save time and costs in implementation projects.

This page intentionally starts on an odd page, so that it is on the right half of an open book from the readers point of view. This is the reason why the previous page was blank (the previous page is the back side of the cover).

Connect Installation Guide

Printed: 2016 in Espoo, Finland

Publisher

Software Point Oy

Technical Editors

Raimo Oinas

Table of Contents

1 Introduction

1.1 Welcome

Welcome to the Connect Installation Guide.

This document contains the instructions of how to install the Connect software.

Program and document info:

Connect Version:	8.0.0
Document version:	2.1
Published:	2/23/2016
Copyright:	© 2016 Software Point Oy

1.2 Contents

This document contains the installation instructions of the Connect software with a brief discussion about the architecture and technical structure of the software.

This overview on the properties of Connect in shown chapter Getting Started.

The installation instructions are given in the chapter Installation.

1.2.1 Related documents

Information about the usage of the Connect is presented in the document "*Connect Web User's Guide*".

1.3 What is Connect?

Connect is a stand alone java application that use some standard components.
All of these components are included in the installation package and for those has Software Point Oy licenses to deliver.
The components are:

SerialPort, http://serialio.com

Java Service Wrapper, http://wrapper.tanukisoftware.org

Apache Derby, http://db.apache.org/derby/

Cajo Protocol, https://cajo.dev.java.net/

With Connect one can interface instruments and systems with a LIMS.
Instrument and system interfaces developed with Connect can be reused and can thus save time and costs in implementation projects. Connect is utilized in implementation projects for industrial as well as medical LIMS systems.

Connect is a tool with a generic framework called the Interface Core Solution API (ICS API) that is especially designed for instrument and system interfaces. With this generic framework it offers a reusable way to support all kinds of instrument and system interface connections to LIMS.
For the LIMS connection, Connect uses a generic messaging framework called Common Data Format (CDF).

1.3.1 Instrument and system interface support

Connect supports instrument and system connections based on file transfer, serial port or TCP/IP (client and server) communication.

Examples of supported instrument types:
- Unidirectional instruments, where the instrument sends results to the instrument interface.
- Bidirectional instruments (in Batch mode), where the instrument receives a work list from Connect. Based on the work list the instrument executes the required analyses and sends the results back to Connect.
- Bidirectional instruments (in Query mode), where the instrument itself identifies the samples by a barcode, and then asks Connect for the analyses to be performed on the samples, Based on that the instrument executes the required analyses and sends the results back to Connect.

Examples of supported instrument interface low level protocols:
- ASTM 2.0
- Several instrument specific protocols (e.g. for Sysmex XT2000i, Olympus PK7300, BCS Coagulation Station instruments)

Examples of supported system interfaces:
- HL7 v2.3 (Message type)
- MLLP (Minimal Low Level Protocol)

Connect handles instrument and system connections differently based on their category. The categories as described in the table below:

Category:	Description:
Cat I	Simple instruments producing one or more results for unidentified samples and parameters , e.g. balances and pH-meters.
Cat II	Instruments with one-way communication producing data containing information on sample ID, test ID and results.
Cat III	Instruments with two-way communication: uploading sample and test-information from LIMS to instrument, and process data from instrument containing sample ID, test ID and results.
Cat IV	Complex and unique instruments and system integration.

These categories will be referred to in other parts of the document. The term "simple instruments" in any part of the document refers to Category I instruments and system, and the term "complex instruments" refers to all other categories, i.e. Category II, III and IV instruments and systems.

For "Category I" (simple) instruments (e.g. balances and pH meters) only parameter results without sample identification are returned to Connect and parsed over to the LIMS. In normal case the user has required the result from the LIMS data entry page and will receive the results in corresponding cells.

1.3.2 Supported LIMS

Connect version 8 and newer supports LABVANTAGE LIMS version 8 and newer.

1.3.2.1 Historical Version Matrix

The following table shows cross reference about the supported versions of LabVantage LIMS and LabVantage Connect. * means that the combination of LabVantage LIMS version and LabVantage Connect version are compatible.

LabVantage \ Connect	3.2	4.0 - 4.2.2	4.3 - 4.4	7.0-7.2	8.0
R4.7 - R5.1	*				
R5.2		*			
6 - 6.0.2		*	*		
7				*	
8					*

1.3.3 Interface core solution (ICS API)

Connect includes a hierarchical framework API that is called the Interface Core Solution (ICS). The API consists of generic layers for database connection, communication and driver. The advantage of this kind of hierarchical structure is that it enables a very versatile support both for all kinds of instruments and systems and for all kinds of

message types received from various instruments or systems.

When implementing new kinds of instrument or system interfaces, all the layers can be reused in implementing the interface to Connect.

1.3.4 Common data format (CDF)

Common Data Format (CDF) implementation package in Connect provides a reusable way to connect the Connect system with LIMS. With the CDF, the communication between LIMS and Connect is handled with the standard CDF messages, and new instrument connections only need to handle the communication between CDF in Connect and the connected instrument.

CDF handles the data transfer between Connect temporary database and LIMS. Custom instrument instances only need to insert data into the temporary tables, and further processing is delegated to the CDF common instance in a general way. The temporary table structure is always fixed when using Common Data Format. This makes it possible to implement generic instrument interfaces.

The CDF instance packs data from temporary tables into a common data format and sends it to LIMS, as well as receives data from LIMS in common data format and inserts it into temporary tables. The interface in LIMS has to implement the following features:

1) Retrieve instrument definitions
2) Retrieve work lists
3) Insert results
4) Retrieve ongoing tests

The procedure is called by Connect, using a HTTP Servlet interface.

For LabVantage LIMS, an action will be installed which handles the transferring of results from the InstruResult SDC into LabVantage data sets, i.e. Dataitem SDC as well as cleaning up the InstruResult SDC after successful operation.

1.4 Getting started

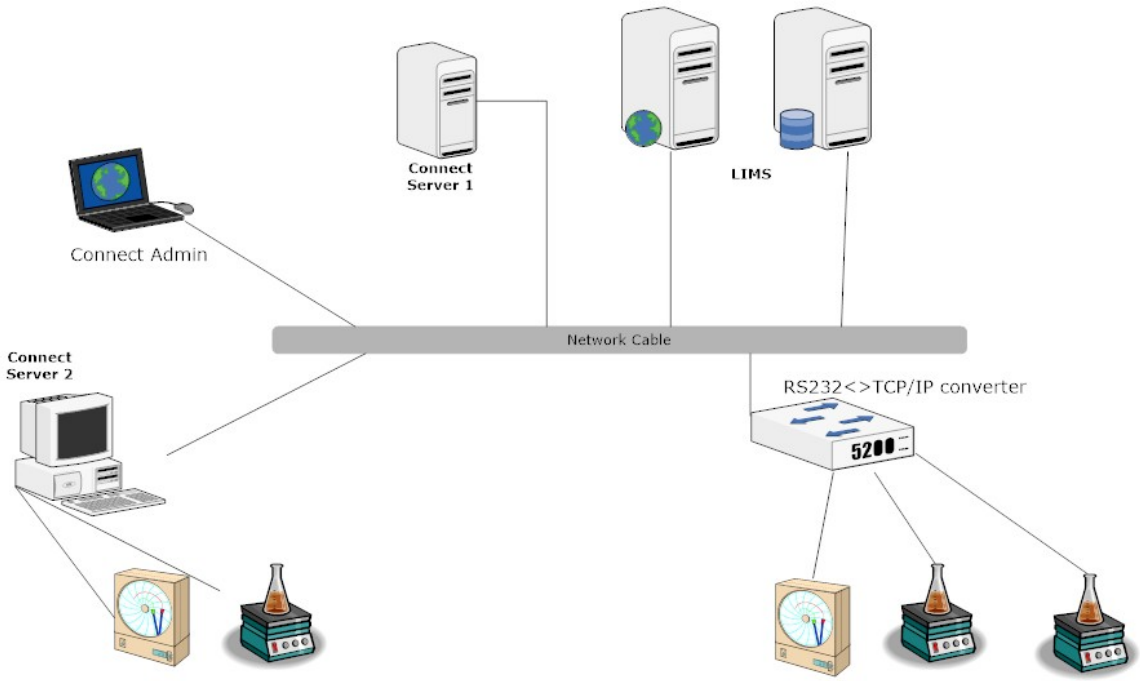
1.4.1 Technical structure

Connect includes a hierarchical framework consisting of generic and configurable layers for database connection, communication and driver parser. The advantage of this kind of hierarchical structure is that it enables a very versatile support for all kinds of instruments (bi- and unidirectional) and all kinds of message types received from various instruments.

All the layers are saved to the instrument library and can be reused in implementing new interfaces that do not exist in the standard Connect library.

1.4.1.1 Environment

Connect is designed to work in following environment:



Connect Application is generally installed as a Windows NT Service or as a Unix/Linux Daemon. The service/daemon holds all the functionality of Connect.

Connect can be configured over a web-page, which by default connects to the Connect Application via HTTP port 8580. Connect web-page is used to setup, change or view the properties of the Connect Application.

The configuration made with the Connect Client is made directly to the running Connect Application, and saved to an xml-file.

1.4.2 Architecture

The Connect installation package installs the standalone Connect Application.

Connect Application can either be run in console mode (usually in the test phase), or it can be installed as a Windows NT Service, or Unix daemon, to the server. The functionality of the Connect is to run ongoing configuration made by user. Also, if configured, it automatically manages instrument interfaces, system interfaces and LIMS connections and restores connections if needed. If user chooses to reboot the server where the Connect is running, the service automatically downloads the configuration made by user after the reboot. Instrument connections and LIMS connection can be set to start automatically after reboot.

Connect web-page is used to manage Connect Application tasks, such as:

- Configuring and managing new instrument or system interfaces

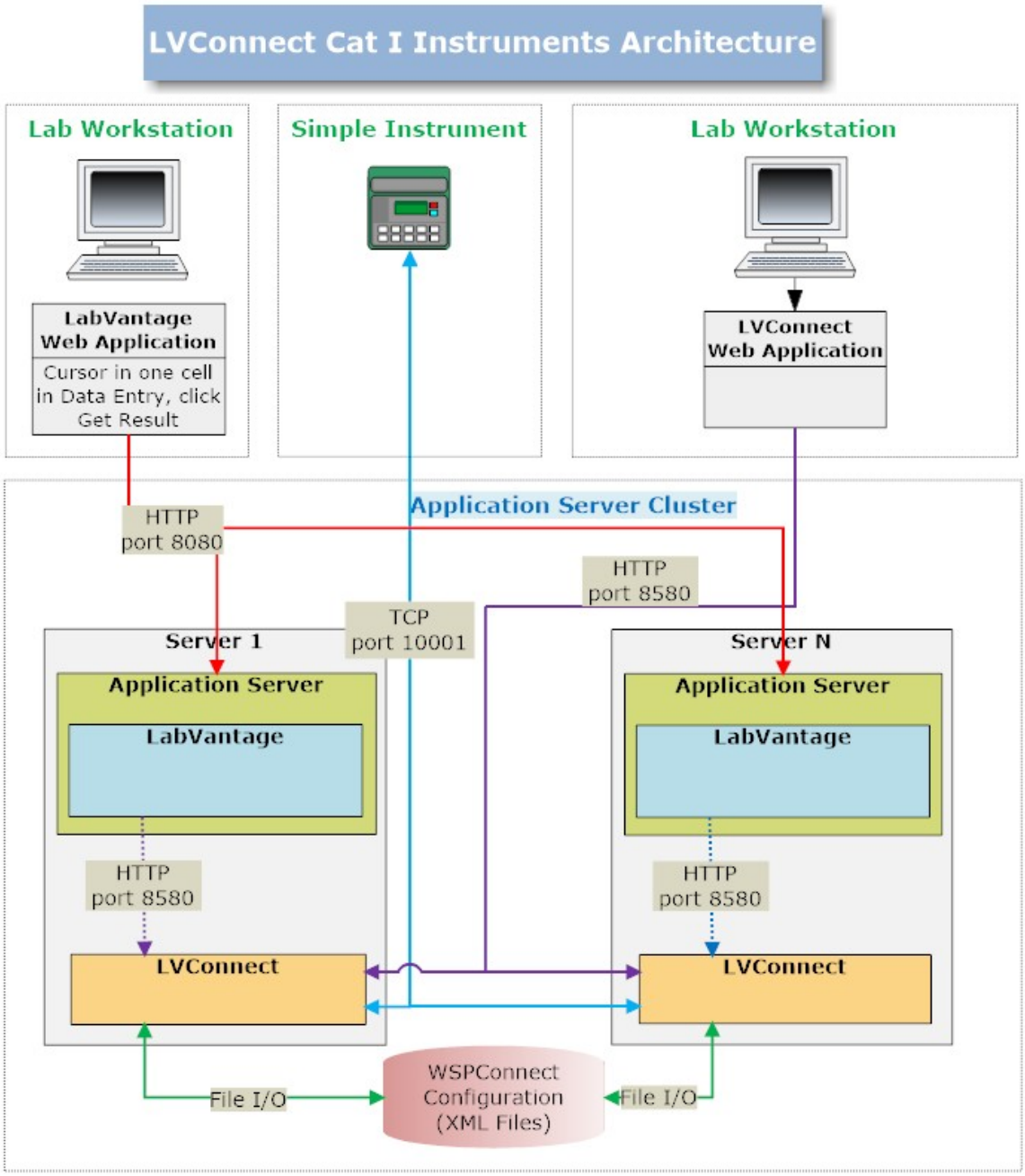
- Configuring and managing new LIMS connections
- Configuring and managing new temporary tables
- Viewing ongoing traffic between Connect and LIMS
- Viewing ongoing traffic between Connect and instrument or system interface
- Sending test messages to instrument or system interface via terminal functionality.
- Managing Connect users and user rights as follows:
  - System administrator, who controls the whole Connect installation
  - Power user, who can change configurations but who is not allowed to add or delete any configuration options
  - User, who only has rights to view and restart connections to other systems and instruments

Connect web-page can be accessed directly with the browser, and also from within LabVantage LIMS.

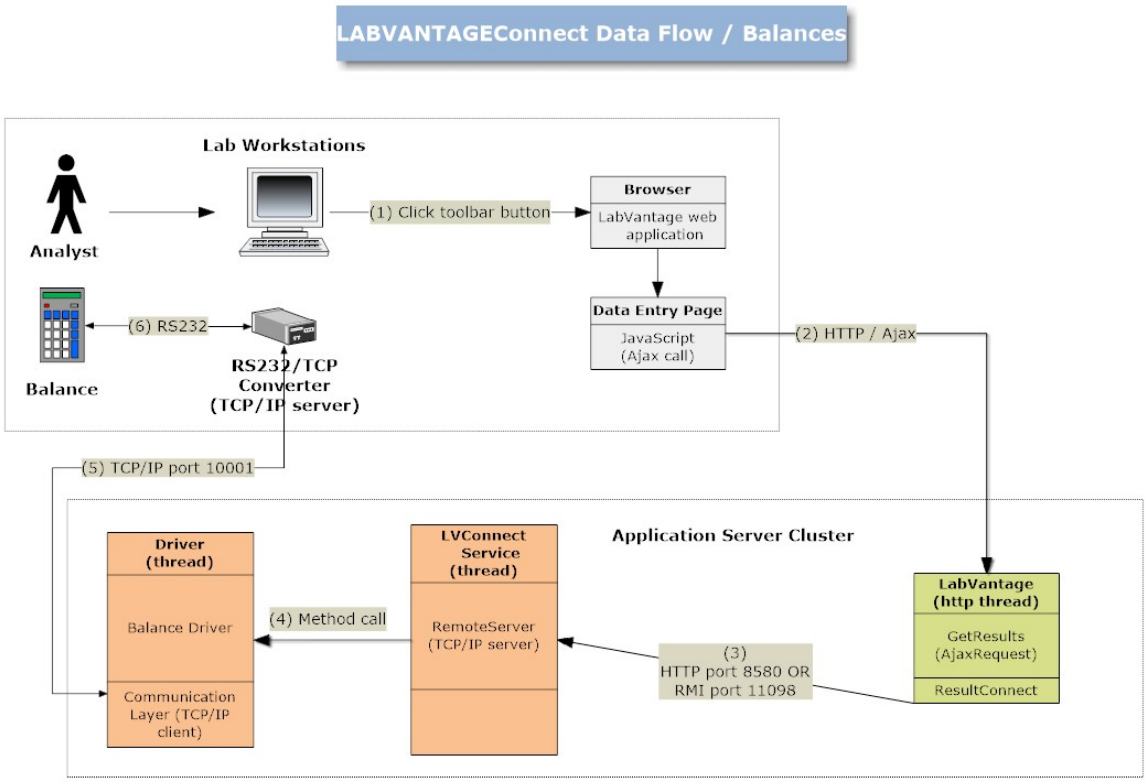
Connect architecture also depends on what kind of instruments will be connected to the system. Simple architecture can be used if only "*Category I*" instruments will be connected. More complex architecture is required only if all kinds of instruments will be connected to the system.

1.4.2.1 For Category I instruments

The picture below shows the technical architecture when connecting to "Category I" (*simple*) instruments that only respond with one result or parameters and result without any sample identification.



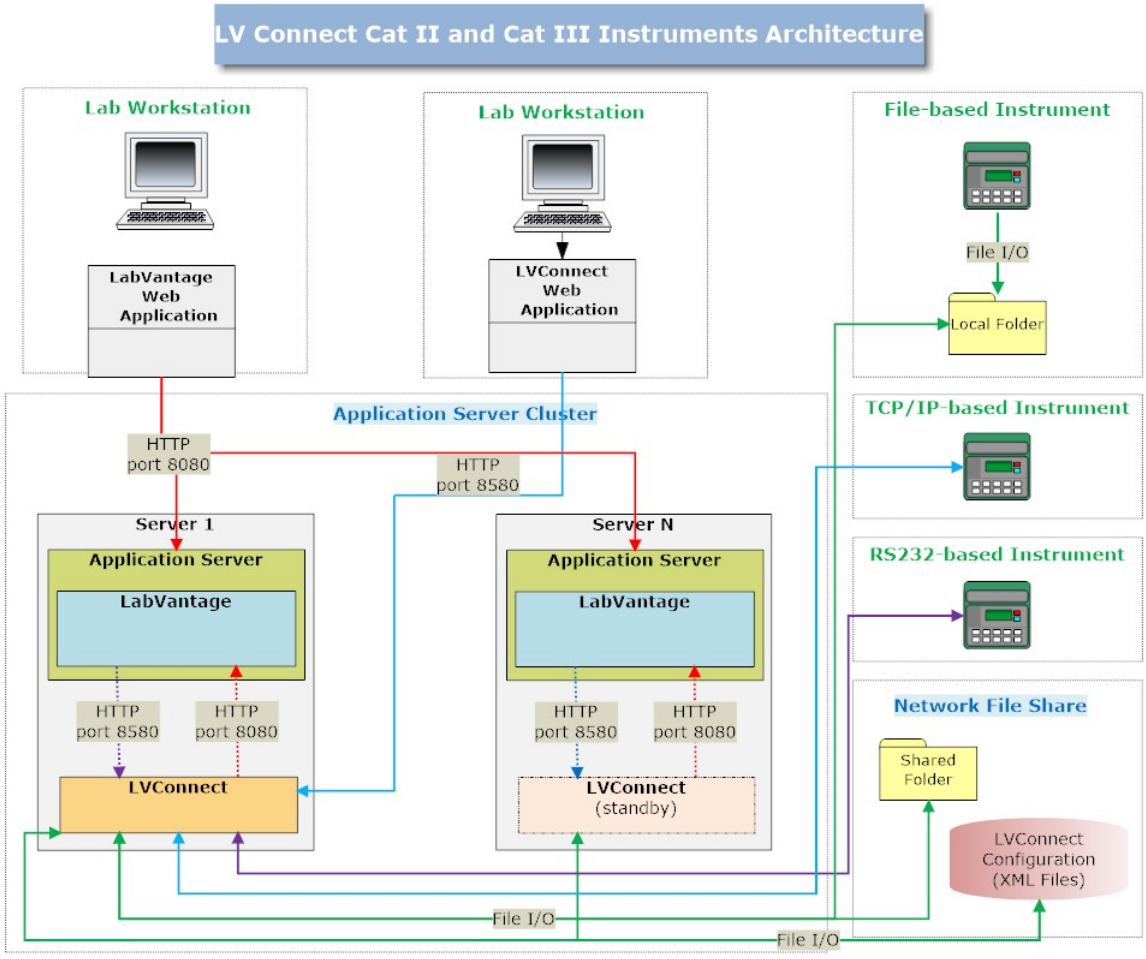
The picture below shows the data flow for how a result can be requested from a balance and return to LabVantage.

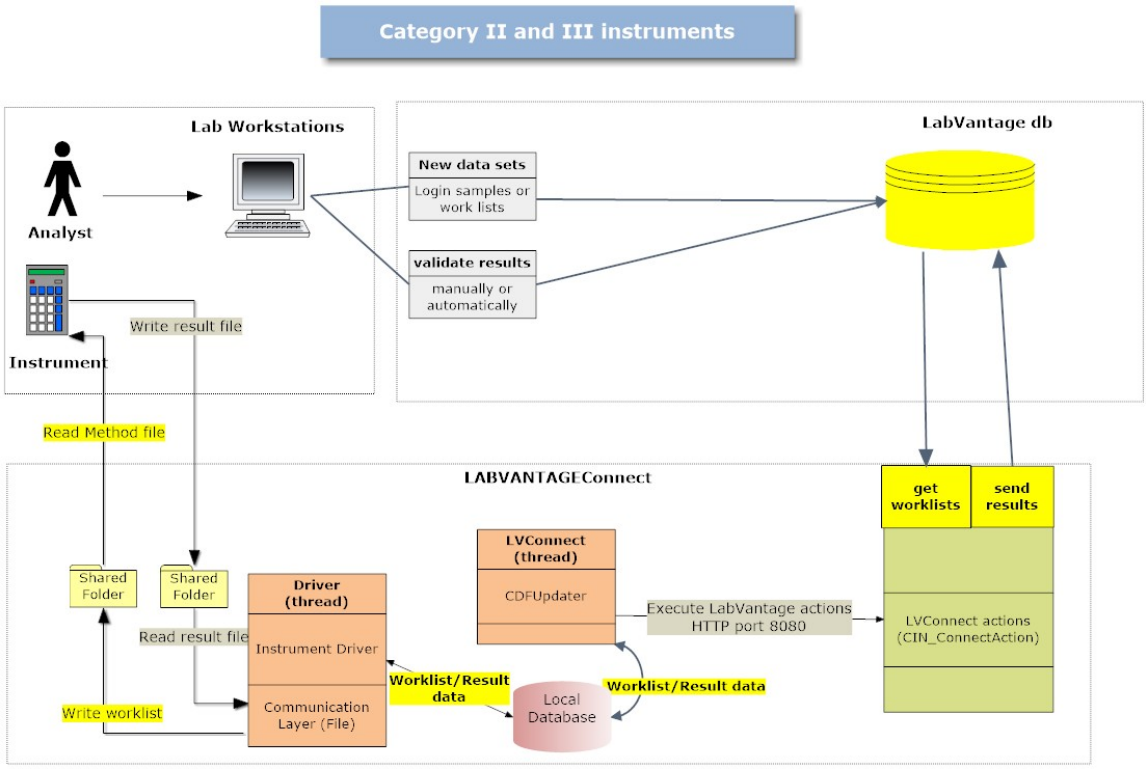


1.4.2.2 For all instruments

The picture below shows the technical architecture when connecting all kinds of instruments.

File-based, TCP/IP-based and RS232-based instruments communicate with Sample ID, Parameter ID one or two way with Connect.





1.4.3 Connections

Connect operates between the instruments / systems and the LIMS system.

1.4.3.1 To instruments and systems

Connect supports two-way communication using TCP/IP (client or server), Serial port, and file system. Connection properties can be easily configured using the Connect web-application.

Connections are handled in Connect as instances. Every instance has its own set of parsers, according to the specifics of the data handled with that instance, and handle moving data between the instrument, Connects own database, and LIMS. There is generally one instance per instrument in Connect.

There can be several instances in Connect relating to one system interface, every instance in this case handles some isolated part of the system interface.

"*Category I*" instruments, like balances and pH meters, that do not deliver anything else than a discrete result shall be implemented with one LIMS Connection but it's not needed to be connected. They only listen to a request from a LIMS user and respond with the result. Every other instance is connected to (at least one) LIMS Connection.

1.4.3.2 To LABVANTAGE LIMS

Connect natively supports connection to the LABVANTAGE LIMS via the WSP Servlet module, which utilizes the standard LABVANTAGE API.

For complex instruments (deliver Sample id, parameter code and results) Connect offers functions like fetching instrument parameters from LABVANTAGE and sending results to LABVANTAGE.

With complex instruments the connections to LABVANTAGE are handled within the LIMS Connection instance running in the Connect service. For the LIMS Connection you need to define the URL to the WSP Servlet module.

For simple instruments Connect offers direct result fetching from the instrument and presenting it on standard LABVANTAGE Data Entry pages.

With "*Category I*" instruments the connection to LABVANTAGE is made by Ajax call from the web page to the LABVANTAGE application server, and from there to Connect with the RMI call (cajo) or HTTP-call. For the LIMS Connection you need to define the Connect host and RMI port, or the URL.

Connect can also work in offline mode, which is useful if the connection to the LIMS database is temporarily broken. During breakdown Connect can still save all the results or requests it has received from instruments and systems into its own lightweight database, and will automatically update results to the LIMS database as soon as the connection is restored.

2 Installation

Connect installation consists of installing the Connect web-client and Connect service and configuring the service to be executed in Windows service or Unix/Linux daemon and installing the Connect interface to LABVANTAGE with LABVANTAGE console. Installing Connect interface will modify (labvantage)ear package and import Connect's pages, actions and so on to LABVANTAGE Lims.

2.1 Connect Application

Connect web GUI plus Connect service that runs in background is referred as Connect Application.

2.1.1 System requirements

Connect requires following elements:

Operating systems:

- Microsoft Windows 32/64-bit, e.g. 7/8/10/2008/2012 server

- Unix 32/64 bit, e.g. Oracle Sparc
- Linux 32/64 bit, e.g. Red Hat Enterprise Linux

Java:
- JRE version 7 or newer or JDK version 7 or newer if the Connect server is supposed to handle more than 500 instruments on one server. JRE/JDK can be downloaded from https://www.oracle.com/java
- SerialPort -implementation requires Java version 8

Disc space:
~ 250 MB

Web browsers:
- Microsoft Internet Explorer 10 and 11
- Google Chrome 37.0 or higher
- Apple Safari 6

2.1.2 Installed components

Connect installers install the following modules / components into your servers and client workstations:

Location:	Module:	Manufacturer:
Connect server	Connect service	Software Point
- " -	Connect web admin client	Software Point
- " -	Java Service Wrapper	Tanuki Software
- " -	SerialPort module	SerialIO.com
LABVANTAGE application server	WSP servlet	Software Point
- " -	LABVANTAGE connector	Software Point
- " -	LABVANTAGE configuration (e.g. data model, queries, actions, web pages)	Software Point

2.1.3 Running Connect Application installer

The installer for Connect server software, Connect client, and the documentation, is the file *connect-installer.jar.* The same installer is used on all platforms.

The "*connect-installer.jar*" can usually be run by double clicking the file. If that doesn't work, try to type

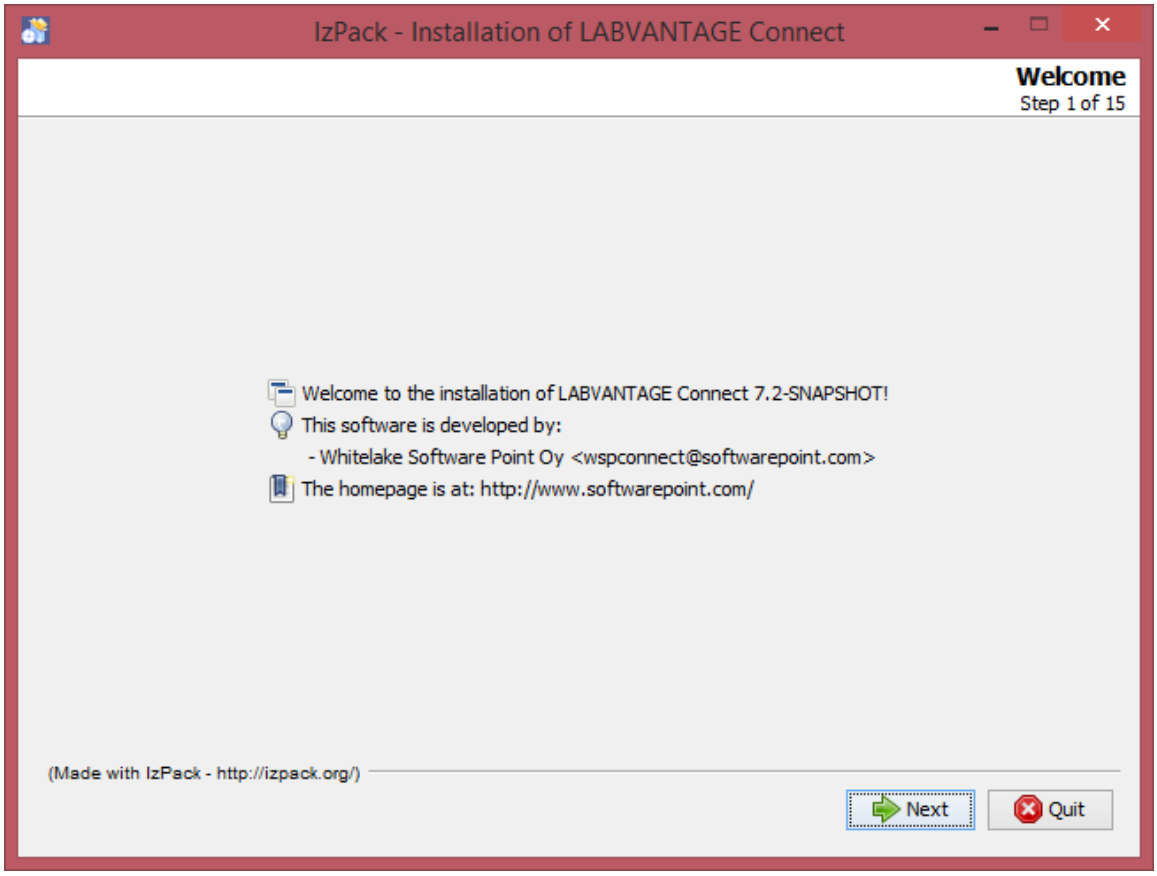
```
java -jar connect-installer.jar
```

at the command prompt.

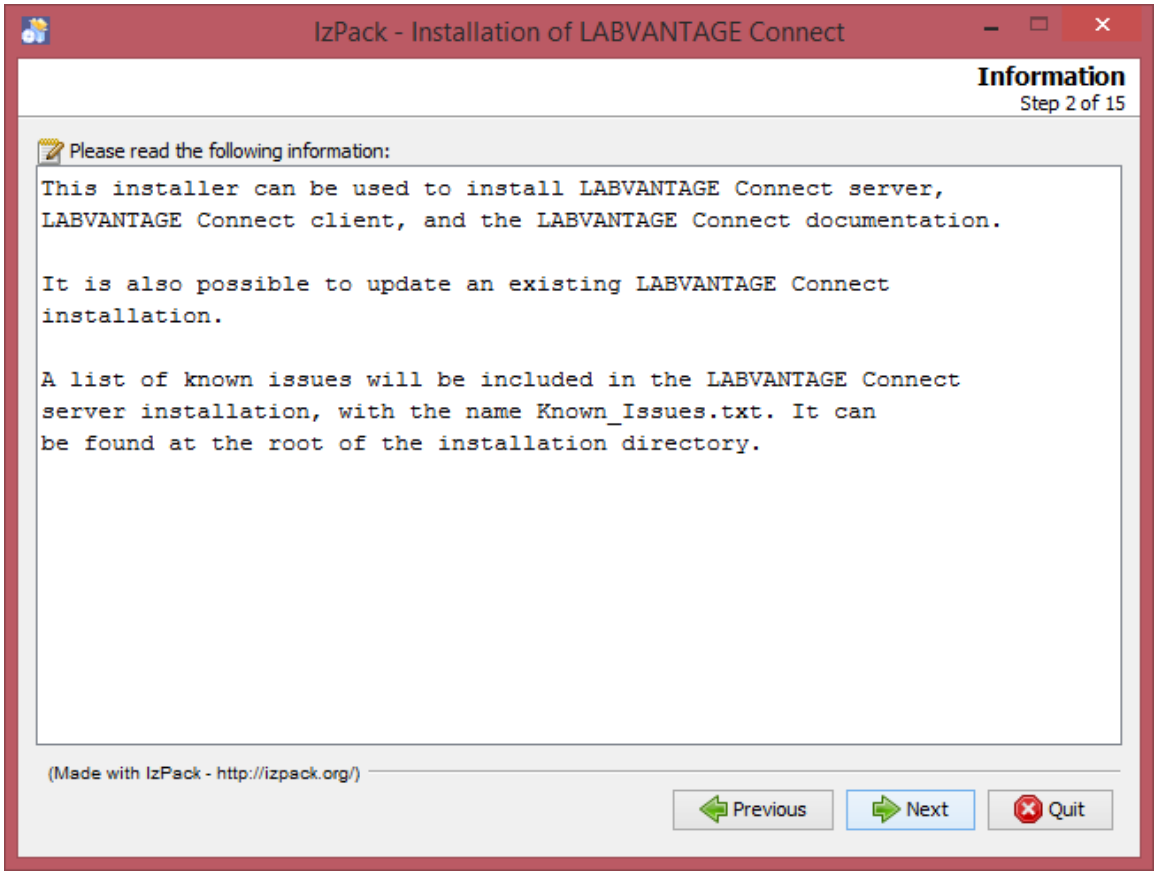
On some Windows versions, the installer will prompt for administrator privileges, which have to be granted. If double-clicking doesn't prompt for administrator privileges, abort the installation and try to start the installer from the command line, it can show the prompt correctly even when double-clicking fails.

The basic installation works identically in Windows and Linux/Unix.

After starting the installation, the Welcome dialog is opened.

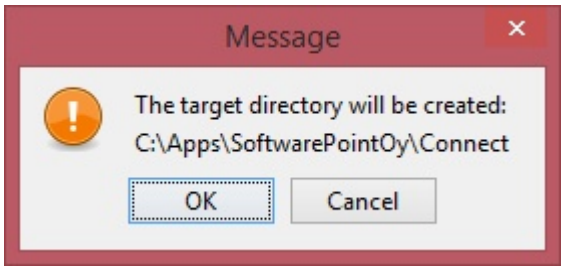


Click Next.

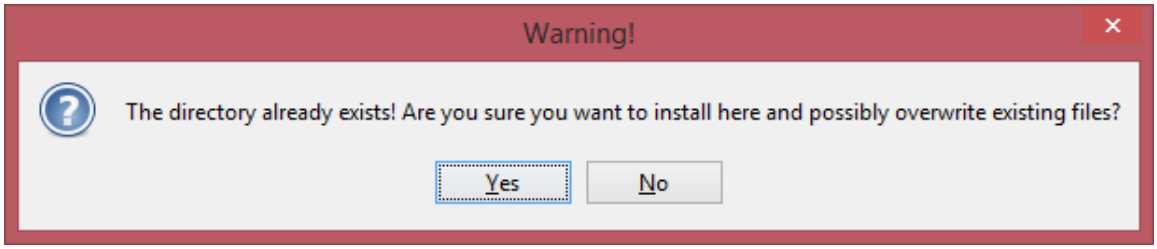


Click next.

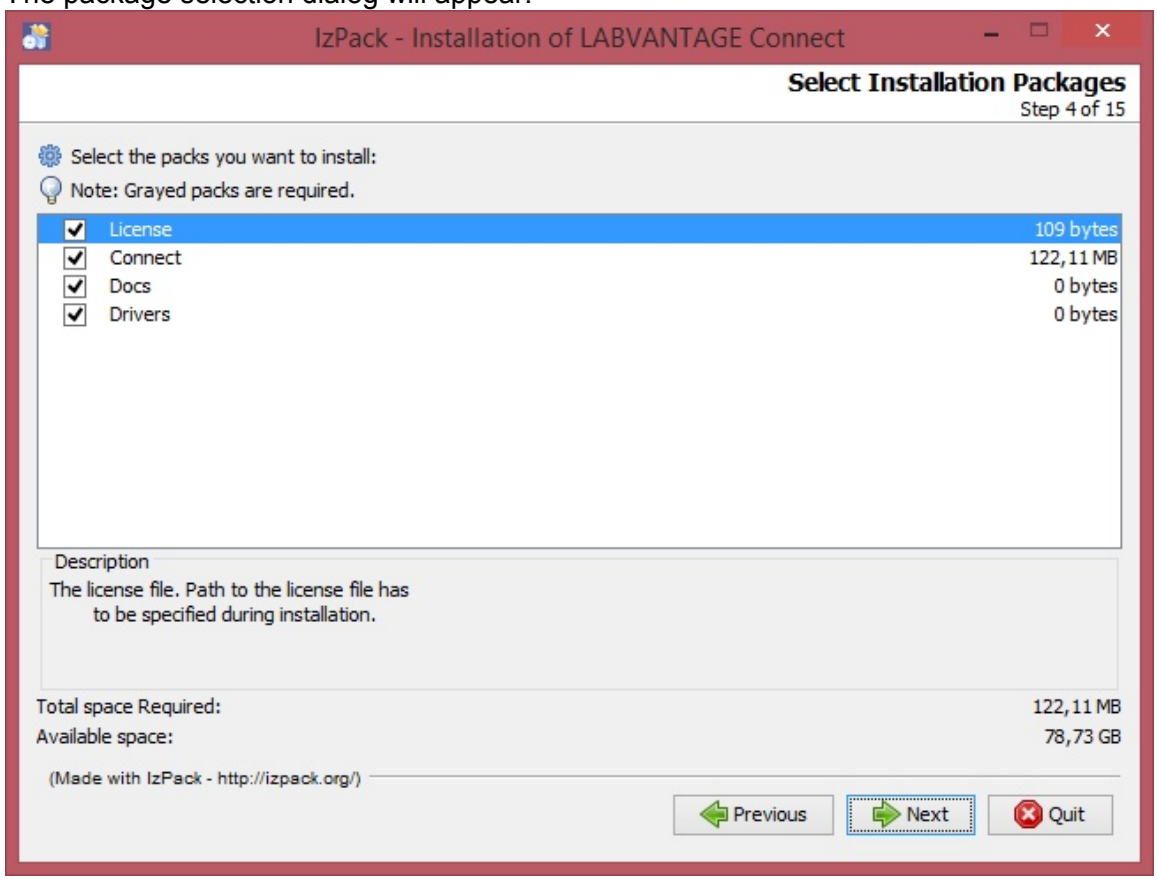
Choose the installation folder, then click Next. If the folder doesn't exist, a warning dialog will be presented. Click OK to continue with the installation.



In case the folder exists, another warning dialog will be presented instead. Click "Yes" to continue.

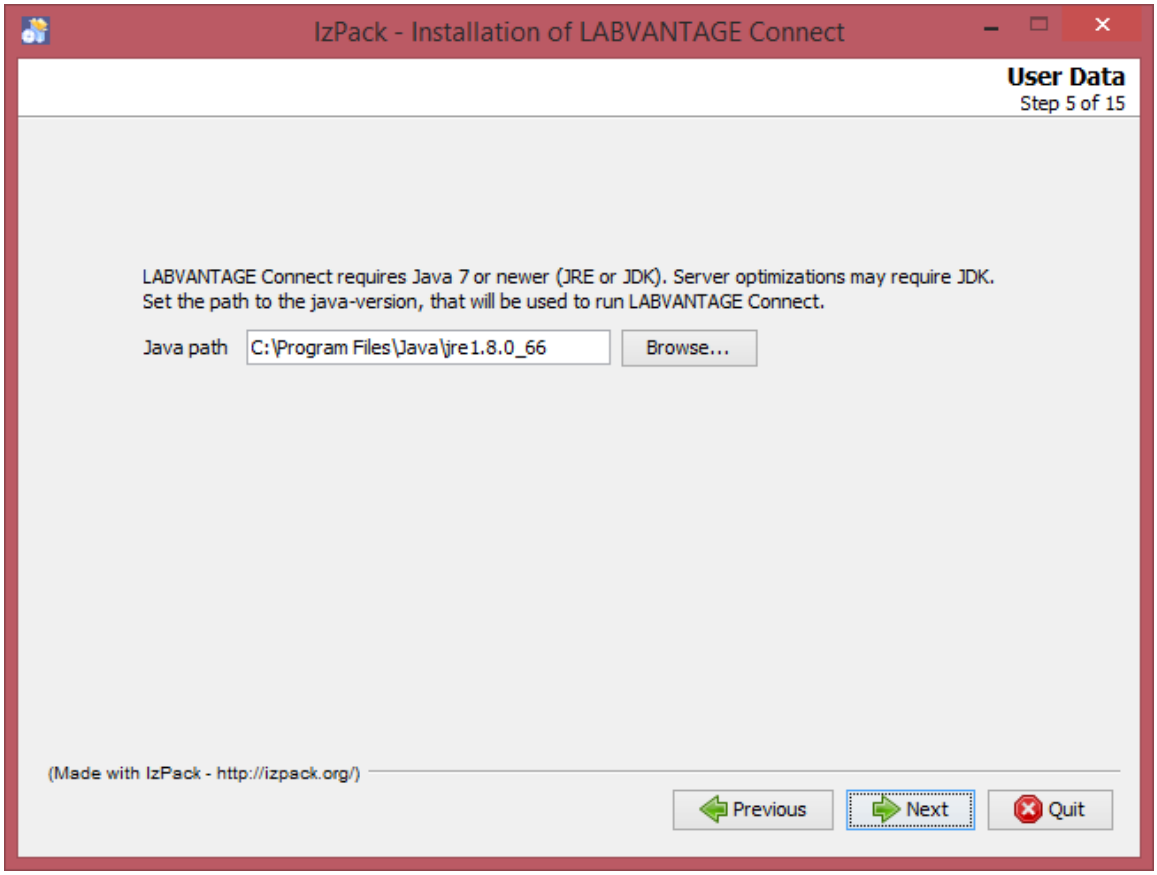


The package selection dialog will appear.



The License and Connect packages is required to run Connect. The Docs package contains the Connect documentation. Drivers-package contains currently implemented drivers and exported instrument configurations.

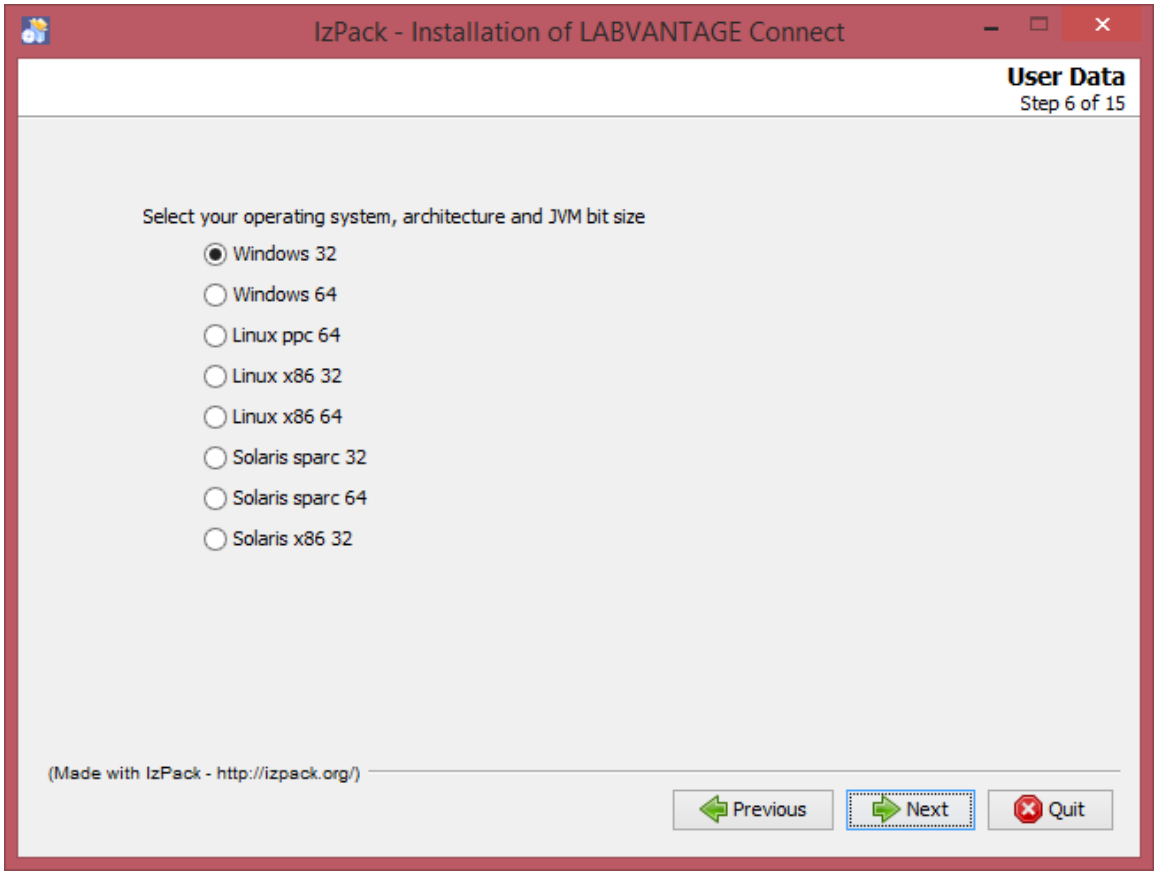
Press Next to continue. If Connect-package was selected for installation, the following dialog will be presented.



System's default java-path is automatically filled in, but you can select the JVM you want to use to run the Connect service (see system requirements).

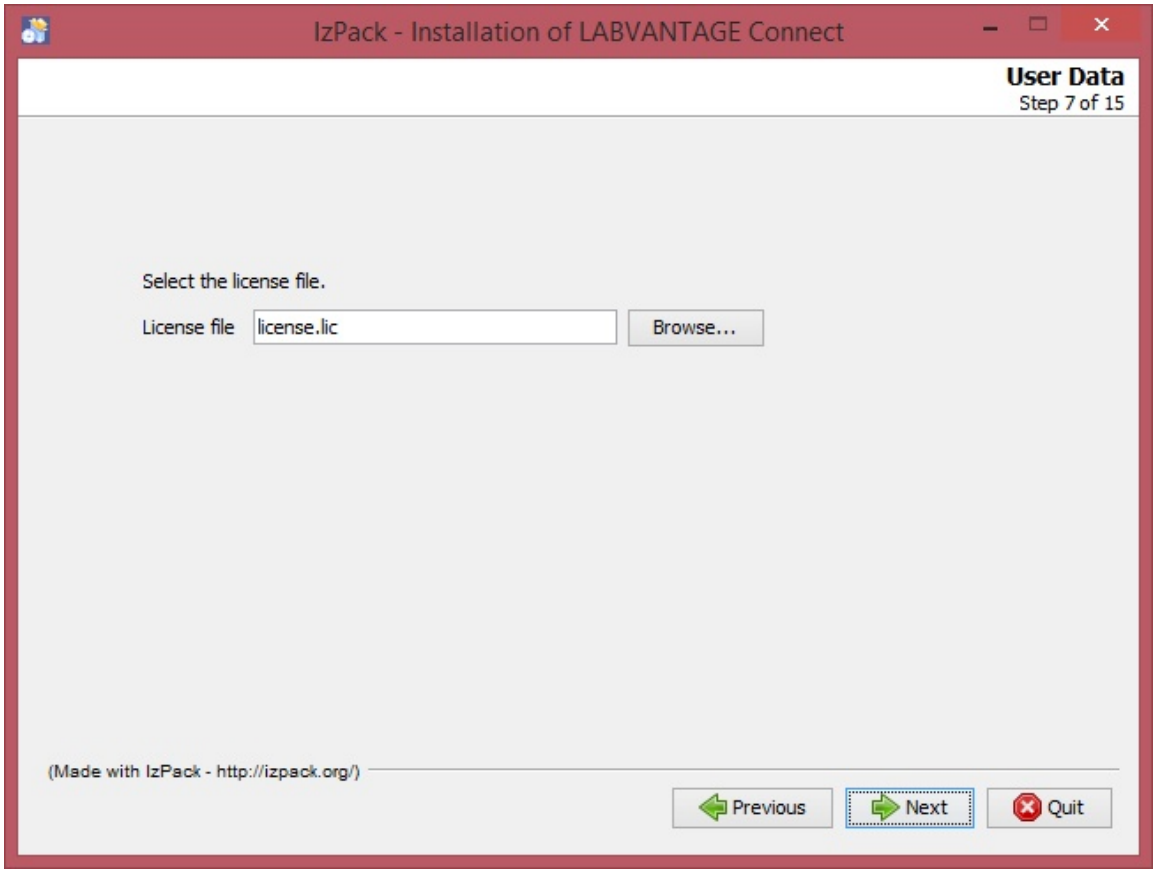
If your system will be running under heavy load, it might be desirable to use JDK instead of JRE. This makes it possible to use `-server` –optimization for Java, providing better performance. See Setting Server Optimization.

After setting the java path, click next to continue. The operating system selection dialog will be opened.

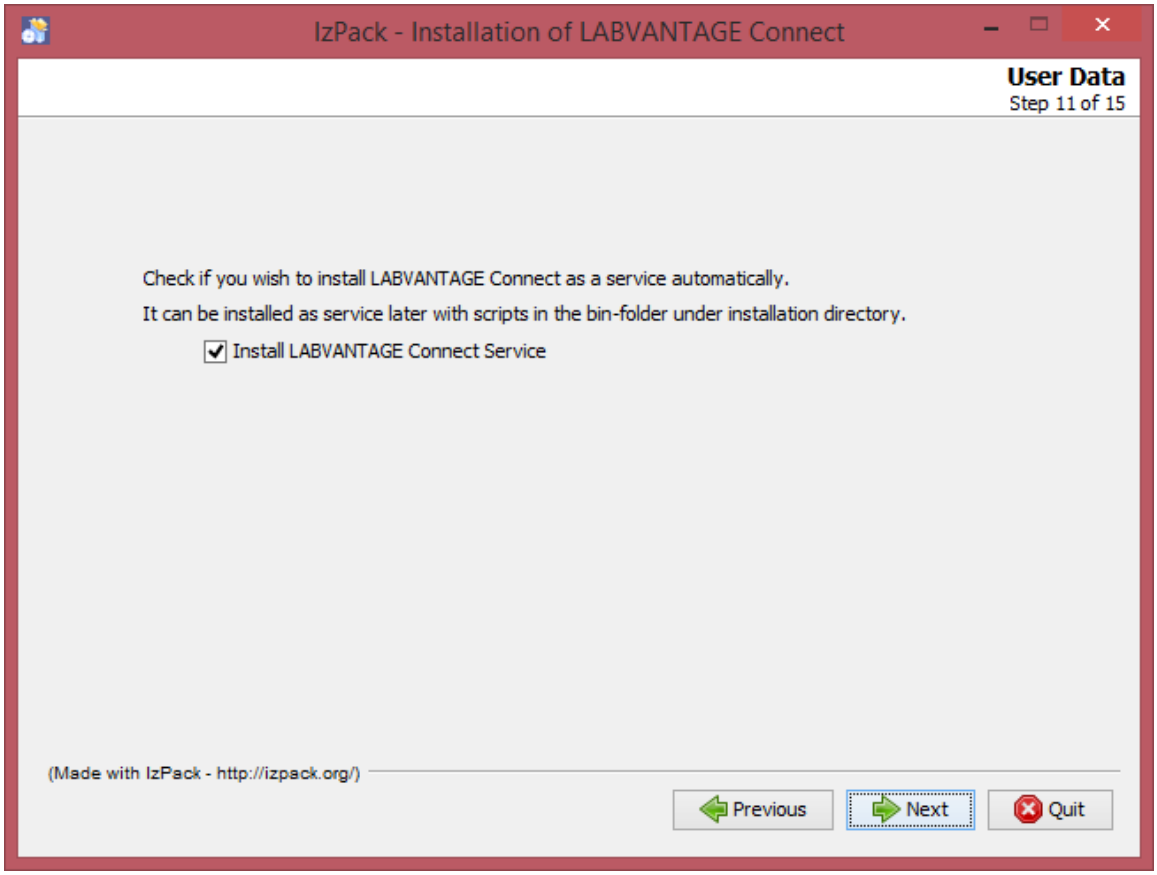


Select the appropriate operating system, and press Next. Note that the bit size (32 vs. 64) refers to the bit size of the referred JVM, not the operating system. 32-bit JVM in a 64-bit operating system is supported, and in that case the 32-bit option should be selected.

If the license-package was selected, the license file selection panel will be shown next.



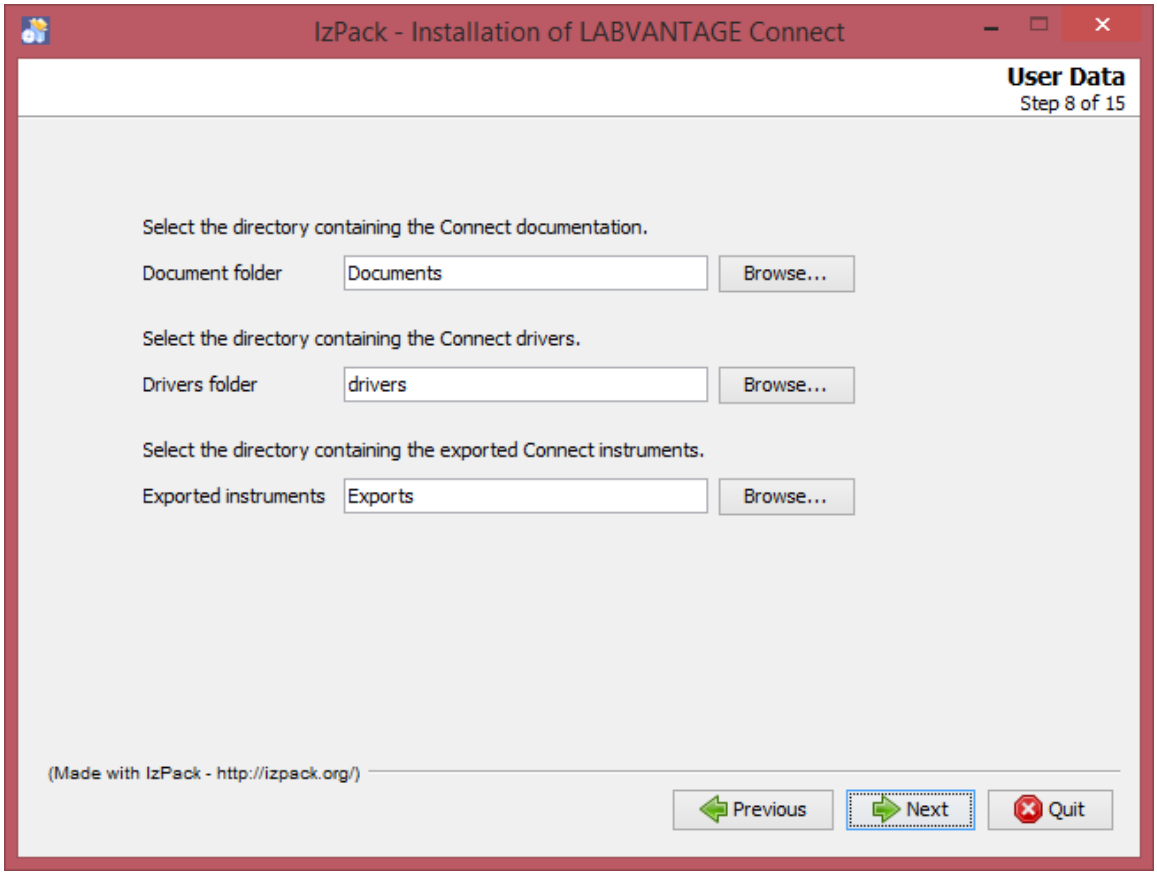
Select the path to the license-file, and press next.



If installing under Windows you can now Uncheck the "*Install LABVANTAGE Connect Service*" checkbox, if you don't want to install Connect as a service at this time. The Service can be installed later by running a script under the installation folder (the script is: *\bin\InstallWSPConnectWrapper-NT.bat*).

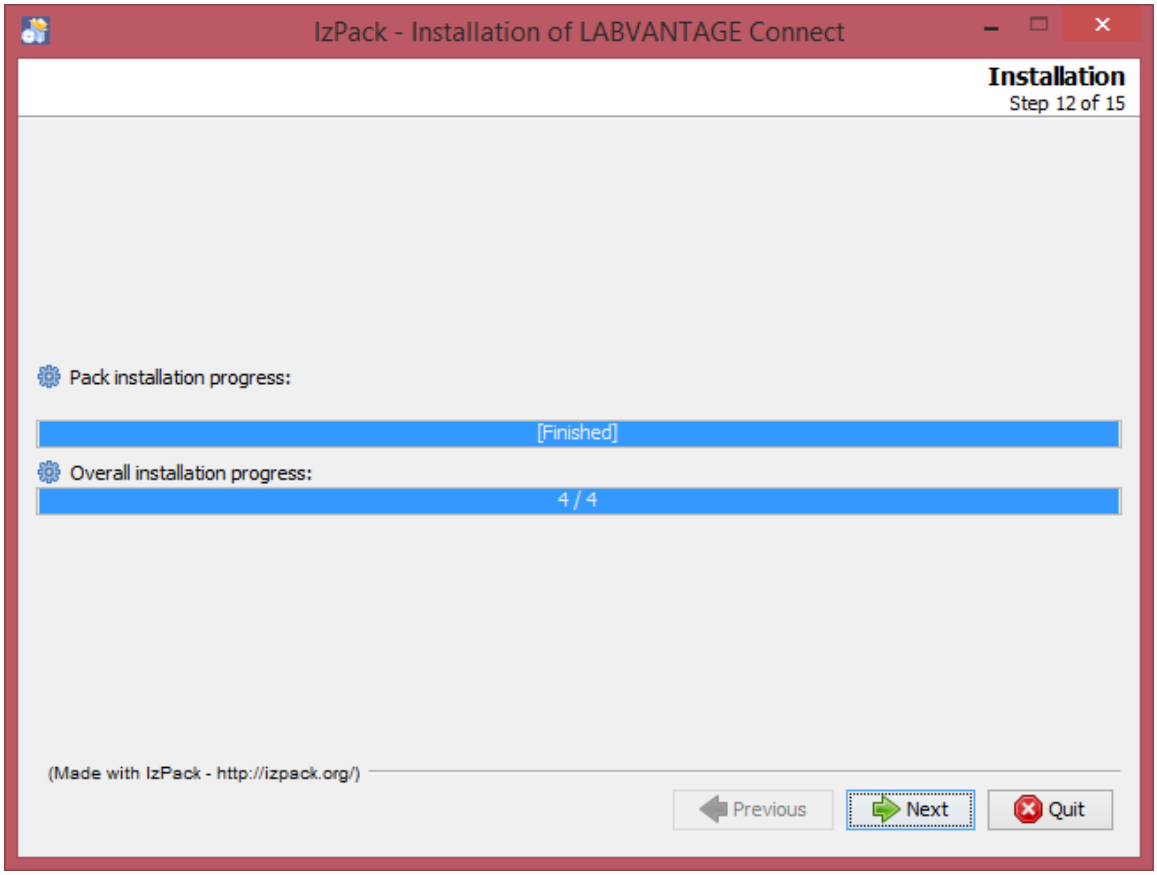
NOTE!: Connect service is not started automatically.

Click Next.



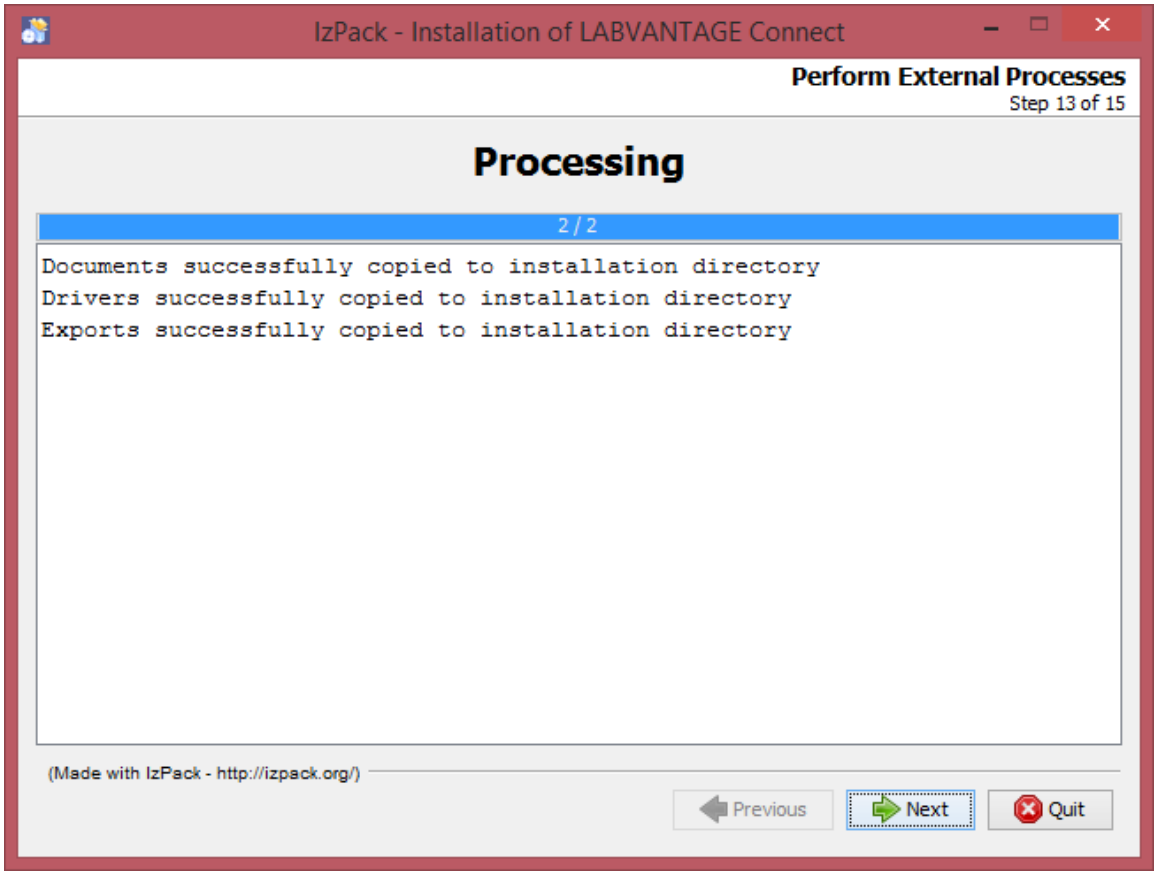
If Docs or Drivers -package was selected for installation, you will now be presented a screen where you can select the directory which contains the documents, drivers and exported instruments. The default directories are correct, if the installation was started from the Connect installation directory, or installation CD.

Click Next.



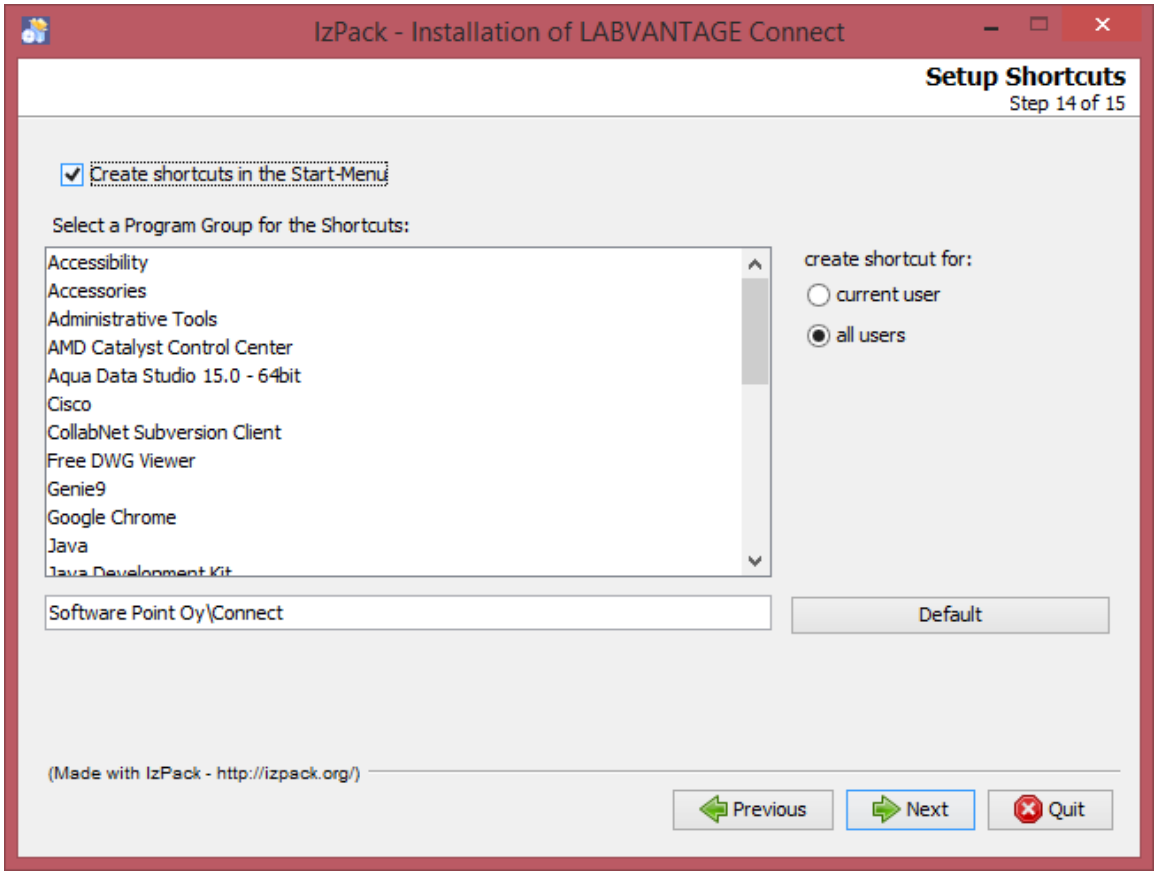
Main application files are copied to the installation directory.

Press Next.



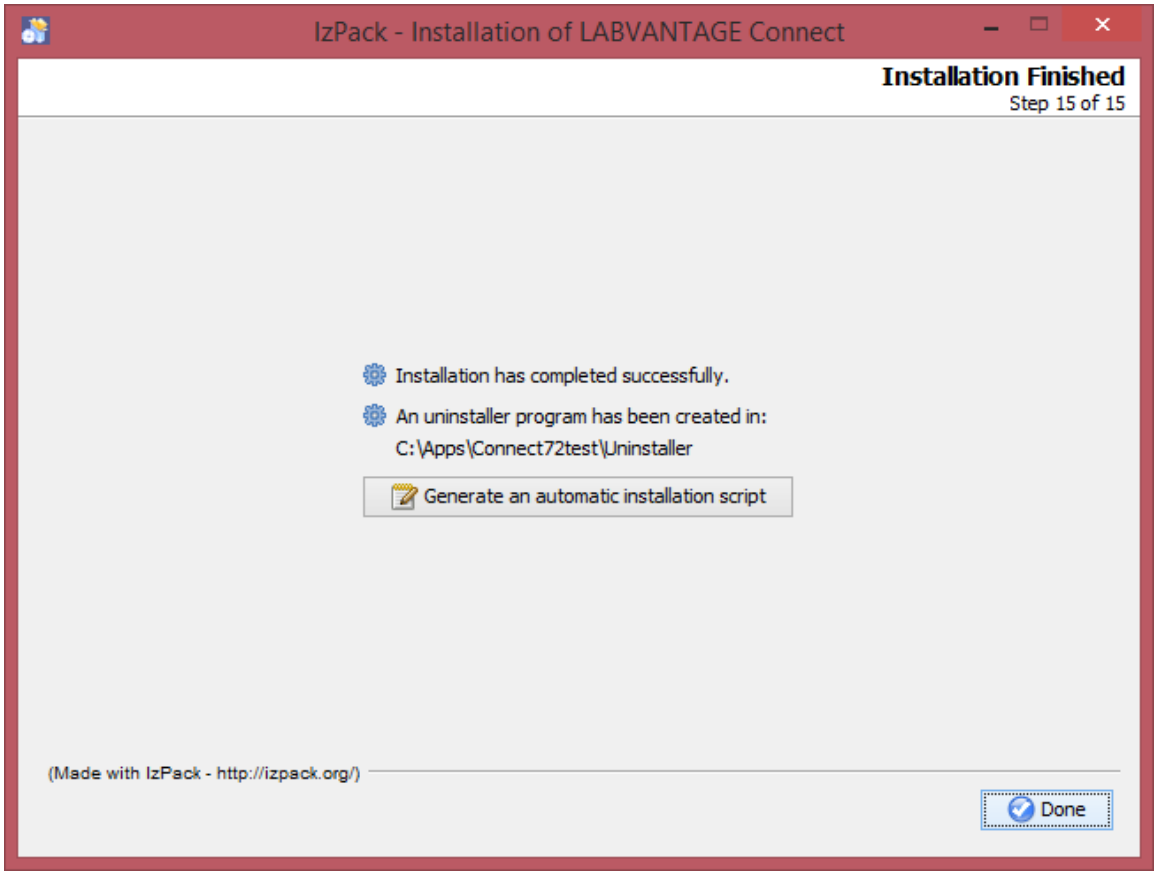
The screen copies the selected Documents, drivers and exported instruments to the installation directory.

Press Next.



Currently it is possible to create shortcuts only in Microsoft Windows.

Click Next.



Installation is now finished.

You can create an automatic installation script, which can be used to replay in another installations, without requiring a graphical user interface. The file must be passed as a parameter to the installer, from the command line.

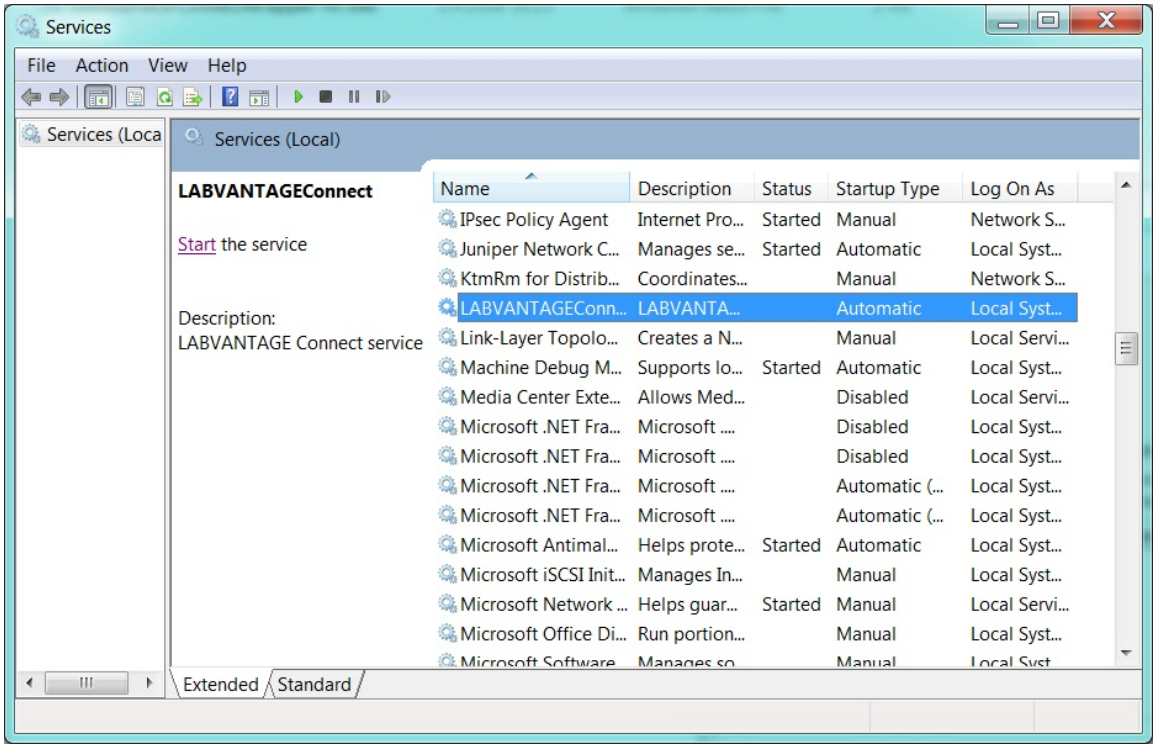
2.2 Starting the Connect Application

This chapter explains how to run the Connect Application in Windows and Unix/Linux environments and how to set up the service as Windows NT service or Unix/Linux daemon.

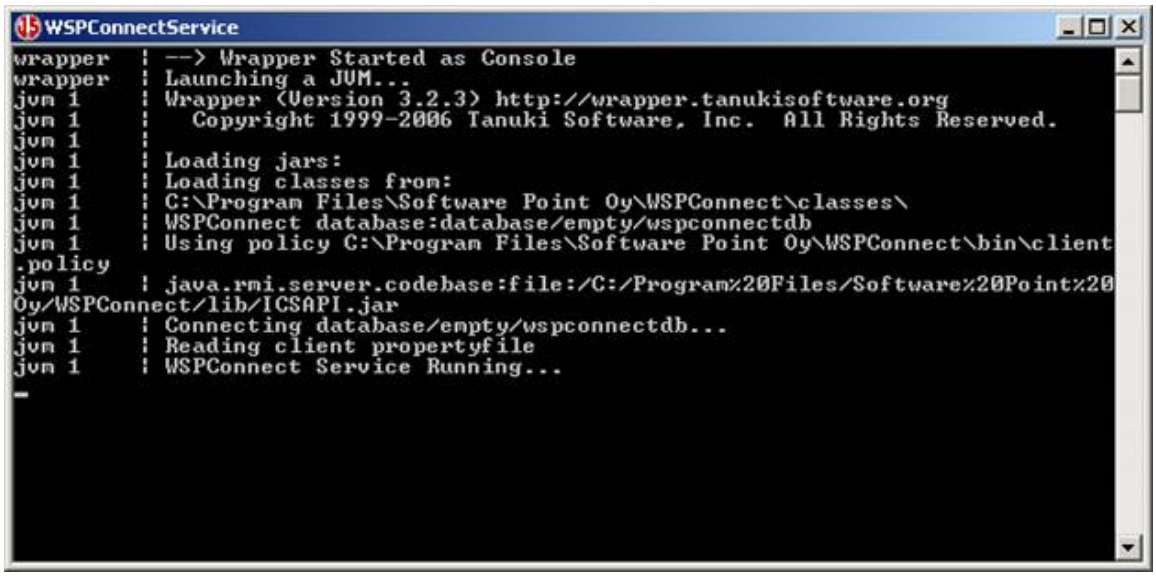
2.2.1 Running Connect in Microsoft Windows

If Connect was installed as a service, it can be seen in the listing of the services running in the computer.

The services can be found, for example, by right clicking “My computer” at desktop and selecting “Manage”. Then double click on “Services and Applications”, and then double click on “Services”. This opens the view shown below. The service is not started automatically after installation, but is set to start automatically when the server is next restarted.



You can also run Connect can also be run in a console mode. This is done by executing the script "*run wspconnect.bat*", located in the installation folder. This will open a window shown below. The Connect service must not be running at the same time, as they occupy the same port. The console can be closed by pressing CTRL+C.



Technically Connect uses the Java Service Wrapper to run it as Windows NT Service. If checking the processes running in the system with Task Manager, for example, Connect is shown as wrapper.exe.

2.2.2 Running Connect in Unix/Linux

Connect execution on Unix/Linux is controlled with the shell script "*wspconnect*". The commands for the script are summarized below.

Command:	Explanation:
./wspconnect console	Start Connect in console mode.
./wspconnect start	Start Connect as a (detached) daemon.
./wspconnect stop	Stop Connect daemon.
./wspconnect status	Show status of the Connect
./wspconnect restart	Restart Connect daemon

Connect that is running in console mode can be stopped by pressing CTRL+C.

If you start Connect with the command

```
./wspconnect start
```

it will run as a detached daemon. This means that Connect service will not shut down when the user logs off, but it will **not** restart when the server is rebooted.

Installing Connect so that it will restart when the system is booted and will stop when the system is shut down depends on the platform. This manual gives instructions for the Sun Solaris and RedHat Linux platforms.

2.2.2.1 Installing Connect as a daemon in Solaris

First, login as root.

Create symbolic link to Connect executable by typing (using the correct installation directory):

```
ln -s /home/tri/SoftwarePointOy/Connect/wspconnect /etc/init.d/wspconnect
```

Test your link in directory /etc/init.d by starting Connect in console mode:

```
./wspconnect console
```

Stop the Connect console.

Set run levels in directory */etc/init.d*:

```
ln -s /etc/init.d/wspconnect /etc/rc0.d/K20wspconnect
ln -s /etc/init.d/wspconnect /etc/rc1.d/K20wspconnect
ln -s /etc/init.d/wspconnect /etc/rc2.d/S20wspconnect
ln -s /etc/init.d/wspconnect /etc/rc3.d/S20wspconnect
```

Reboot to test the installation.

2.2.2.2 Installing Connect as a daemon in RedHat

Services can be also managed through the Service Configuration dialog, but the instructions are given for managing them using the terminal.

First, login as root.

Create symbolic link to Connect executable by typing (using the correct installation directory):

```
ln -s /home/tri/SoftwarePointOy/Connect/wspconnect /etc/init.d/wspconnect
```

Test your link in directory /etc/init.d by starting Connect in console mode:

```
./wspconnect console
```

Stop the Connect console.

Install the service in directory */etc/init.d*:

```
chkconfig wspconnect on
```

Check service status, and try to start and stop Connect.

```
service wspconnect status
service wspconnect stop
service wspconnect start
```

Reboot to test the installation.

2.2.2.3 Removing the daemon

Login as root.

Unregister runlevels in the folder */etc/init.d*:

```
rm /etc/rc0.d/K20wspconnect
rm /etc/rc1.d/K20wspconnect
rm /etc/rc2.d/S20wspconnect
rm /etc/rc3.d/S20wspconnect
```

and remove the symbolic link:

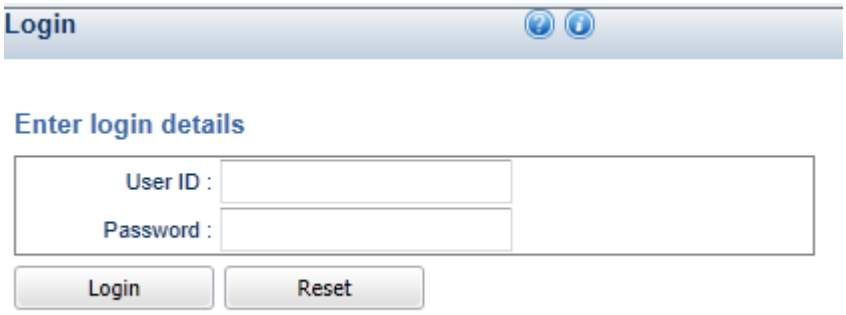
```
rm wspconnect
```

2.2.3 Using the Connect client

To start Connect Client, Connect Service has to be running (either as a service / daemon or in the console mode).

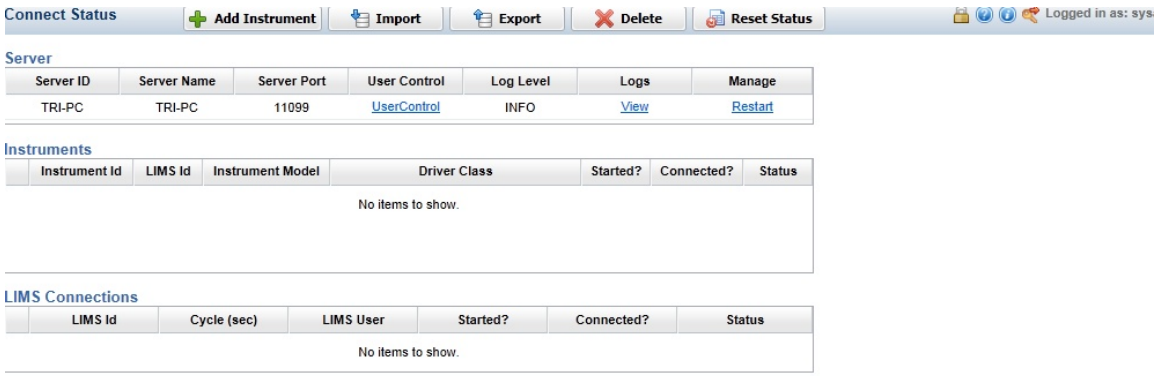
Connect client can be accessed by navigating to address http://servername:8580/ with the browser. Internet Explorer and Chrome are supported.

The following page will open



The default user id that comes with the installation is “*sysadmin*”, with the password “*sysadmin*”. Click “*Login*” to login to the Connect client.

After successful login the main view opens. Now you can import LIMS connection(s) and instrument instances. For details, see the "*Connect Web User's Guide*" manual.



2.3 Connect LABVANTAGE interface

For Connect to work together with LABVANTAGE, Connect (LABVANTAGE) interface needs to be installed. Installation will bring connect functionality to LABVANTAGE, for example, Connect tramline with Connect pages, Connect actions, Connect SDC's, WSPServlet for communicating between LABVANTAGE and Connect, etc. Detail information in following chapters.

2.3.1 What is installed in LABVANTAGE

For LABVANTAGE, the Connect interface installer includes the following jar-packages:

- cajo.jar, which implements the protocol used.
- hostconnector.jar includes 1. Common base-class for all Software Point connector servlets . 2. The WSPServlet which will be installed in LabVantage and receives

communication from Connect. 3. Message structure for communication between Connect and WSPServlet.
- wspconnect-cdfmessage.jar, message-object which is used for communication between LabVantage and Connect for CDF actions
- wspconnect-sapphire.jar, page element for the Connect admin client, actions for CDF called from Connect to receive results/get worklists/tests and to update results from cin_instruresult to sdidataitem, as well as functions for the Category I instruments that poll Connect for results from simple instruments

It includes the following jsp-pages:
- ConnectAdmin.jsp, which is the web page for connecting to the Connect web admin client
- GetResult.jsp, which is called with Ajax and passes the call to the java-class.
- ProgressPopup.jsp, which implements a popup window with which it is possible to interrupt the request for results.
- ProgressPopupGWT.jsp, GWT implementation of progress pop-up.
- ConnectProfile.jsp, which is the web page for setting up Connect auto login credentials

and the following Java Scripts:
- getresult.js, which initiates the call to get results from Cat 1 instrument
- instruname.js, which implements additional behavior to the data entry pages, which makes it possible to store also the instrument name along with the results.
- g_ajaxcallaction.js, helper script to call LABVANTAGE's actions.
- instruResultAction.js, finds out what is selected and calls CIN_InstruResult action.
- importResults.js, which has Validation pop-up (Data Entry page) related code.
- choose_instrument.js, which is used to save instrument id for all datasets visible on the page.
- qcBatchWL.js, which is used to send a worklist from QCBatch-page

2.3.1.1 Connect configurations imported to LABVANTAGE

LABVANTAGE configuration is delivered as LABVANTAGE import files (XML). Following table gives you a basic understanding what Connect configurations are imported to LABVANTAGE.

Configured item:	Package:	Contents:
Data model (SDCs)	CIN_SDCs.zip	New SDCs (CIN_EdiMessage, CIN_InstrumentServer, CIN_InstruResult, CIN_InstruWorkList, CIN_Query, CIN_SysProperty, CIN_statistics) and modified SDCs (Sample, QCBatch, Instrument, LV_InstrumentModel).
Queries	CIN_Queries_Oracle.zip CIN_Queries_SQL_Server.zip	Queries that Connect uses for fetching data from LABVANTAGE database. Own packages for Oracle and SQL Server databases.

Actions	CIN_Actions.zip	Actions for fetching data from and saving data to LABVANTAGE.
Web elements	CIN_Elements.zip	Web page elements for Connect web GUI
Web pages	CIN_WebPages.zip	Web pages for new SDCs
Layout properties	CIN_LayoutProps.zip	Layout property for the Lab Admin Sitemap to include the entries to the new web pages (own Sitemap branch)
Policies	CIN_ConnectPolicy.zip	LVConnect policy node that includes properties to adjust Connect related functionality.
System property	CIN_SysProperties.zip	Includes a few Connect configurations.
LV Queries	LVQueries.zip	Connect queries that are put to LABVANTAGE query table. Used for ex. on Result List page.
Dashboard	CIN_MSS_Dashboard.zip CIN_Oracle_Dashboard.zip	Connect dashboard showing statistics about usage.

2.3.1.2 WSP Servlet

The WSP Servlet is a module that connects external clients (like Connect, QC module) into LABVANTAGE system. To be able to use the connection, the servlet must be installed to your web application. Installing the Connect interface will take care of this.

WSP Servlet uses http and a serialized message object (WSPMessage) for communication between the client and server.

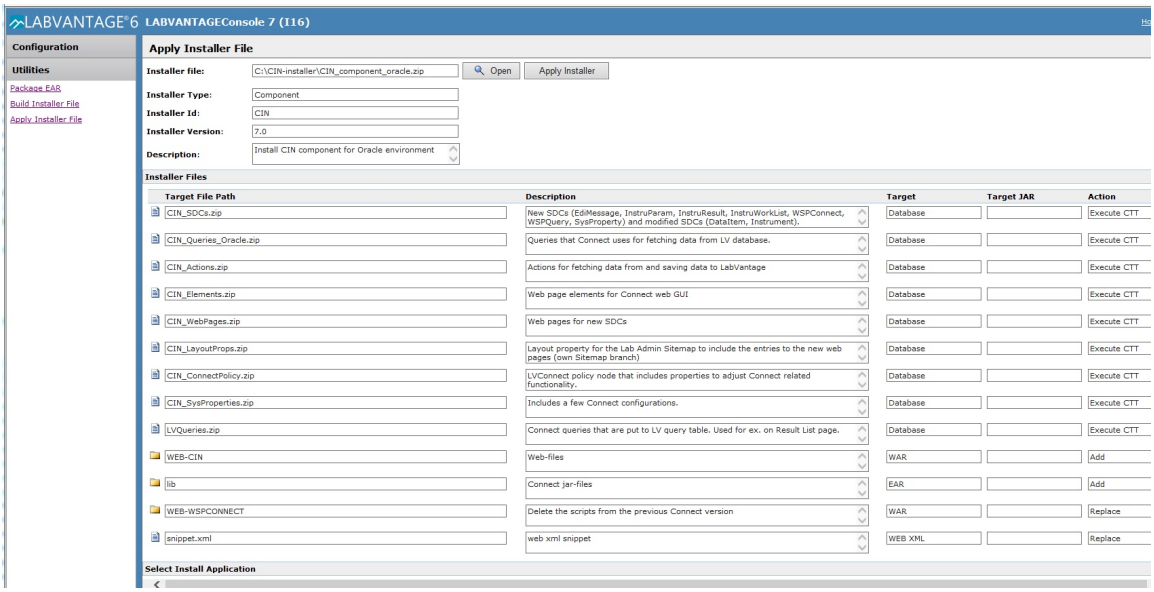
Servlet functionality is packeged in the hostconnector.jar, which includes:

- Serialized message object that is used in the http communication between client and server
- Service classes supporting LABVANTAGE services (connect/disconnect, processAction, etc)

2.3.2 Installing the Connect Interface

The Connect 7.0 (and above) interface is delivered as a zip package that can be installed with LABVANTAGE v7 Console. LABVANTAGE v7 Console is an application delivered with the LABVANTAGE LIMS Package.

Start the installation by opening the Console and navigate to Utilities/Apply Installer File page. Choose a correct installer file according to your environment, package CIN_component_oracle.zip is used if LABVANTAGE database is Oracle, and package CIN_component_sqlserver.zip if LABVANTAGE database is SQL Server. Pressing Apply Installer button will start the installation process.



It takes a few minutes for installer to run. Installer does the following.

1. Modifies the ear package (labvantage.ear). Modifications include adding cajo.jar, hostconnector.jar, wspconnect-cdfmessage.jar and wspconnect-sapphire.jar to ear's lib folder. Adding WEB-CIN folder, which contains for example Connect jsp pages and javascripts to war package. Modifies web.xml to include latest WSPServlet configuration.

2. Imports the Connect configuration to LABVANTAGE. Configuration includes new and modified SDCs, actions and web pages. Detail information about imported configuration is given in chapter Connect configurations imported to LABVANTAGE.

3. Copies the labvantage.ear to deploy directory (production mode).

In the development mode, when the ear is exploded, component installer modifies the deployed ear (the one in JBOSS_HOME\standalone\deployments directory).

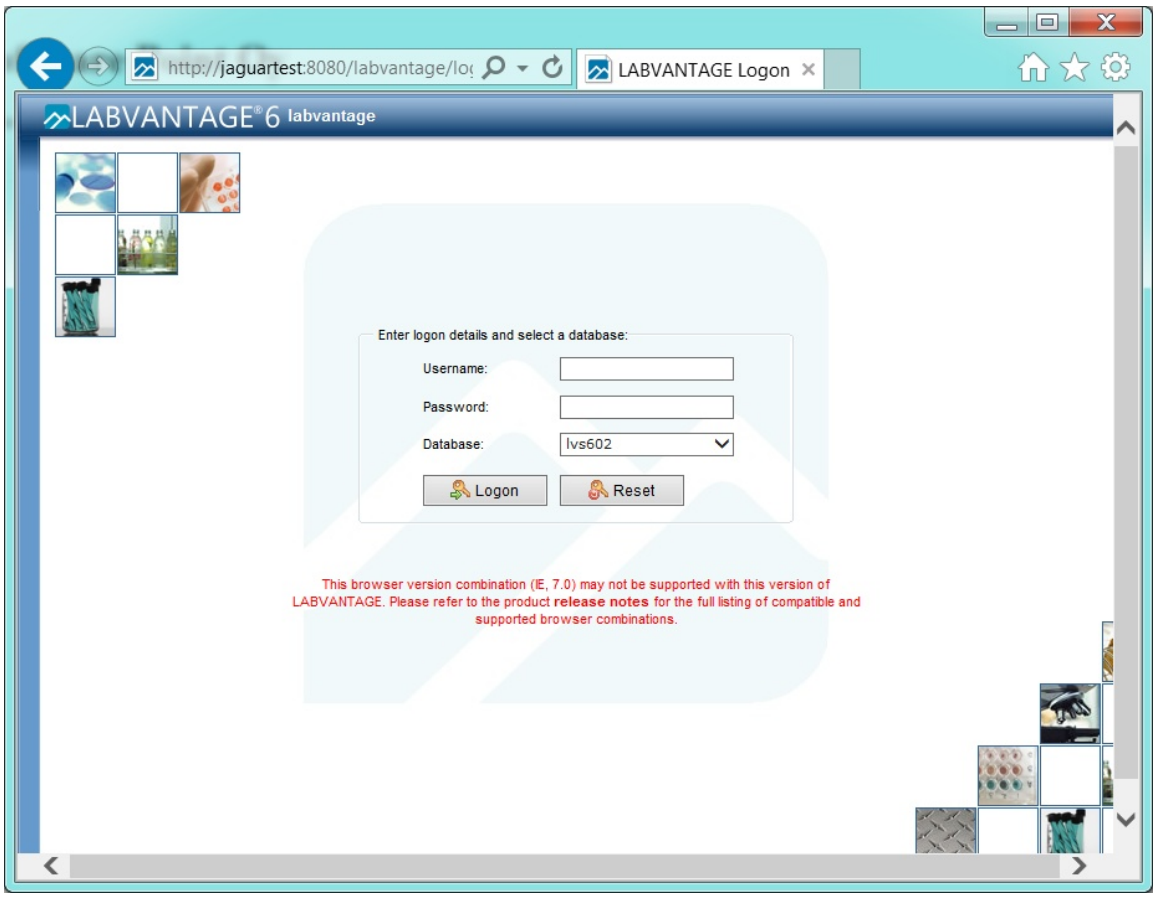
In the production mode, component installer modifies the ear in APPLICATION_HOME/ear directory (i. e. labvantagehome\applications\labvantage\ear).
So If project doesn't use "Managed ear", that is, ear is build using your own processes (rather than those of LABVANTAGE Console), there is a risk that running Connect component installer will cause the project specific ear modifications to be drop out from the ear that is copied to deploy directory.

Notice! If WSPServlet configuration is not in web.xml, installer can not replace it. In that case you can add the WSPServlet configuration to web.xml manually by following instructions on chapter Adding the WSPServlet configuration manually. This feature will hopefully be fixed in some future LABVANTAGE Console version.

2.3.3 Testing the WSPServlet

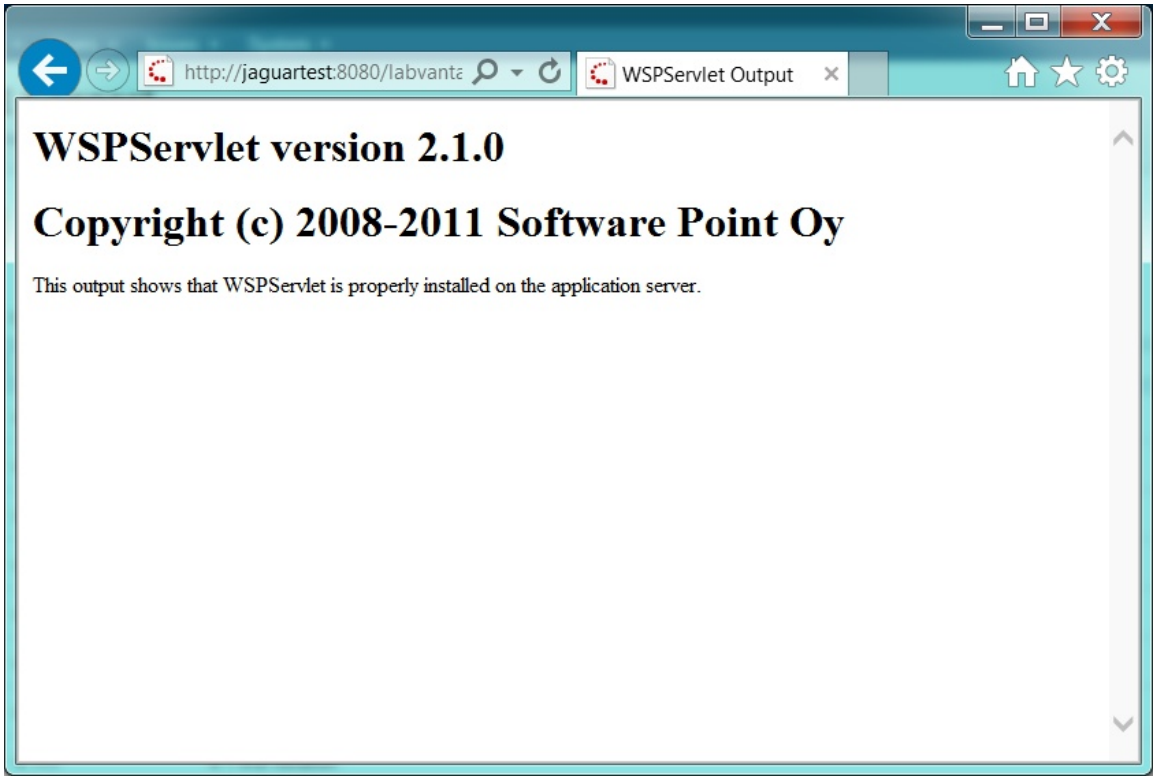
After the installation and application server restart you can test that your WSPServlet has been properly installed. Here is the description on how to do it.

1. Open the web browser and navigate your LABVANTAGE web application http://<servername>:8080/<webappname> like this:



If the LABVANTAGE logon appears, it means that LIMS is running properly.

2. Navigate to the servlet page http://<servername>:8080/<webappname>/wsp like this:



If the WSPServlet info page appears, it means that your WSPServlet is properly installed and configured.

In case you get an error message instead of the WSPServlet info, make sure that your Connector jar packages have been properly installed and the application server has been restarted.

You can also check that web.xml in WEB-INF folder has configuration for WSPServlet. If not, you can add configuration by following steps in chapter Adding WSPServlet configuration manually.

2.3.4 Adding WSPServlet configuration manually

EAR-file can also be modified manually with a zip-tool. Recommended zip-tool is WinRAR, because with WinRAR it is possible to modify files within a zip-package, and unpacking/ packing files is completely avoided. Modifications can be done with any tool that understands zip-packaging, but instructions assume WinRAR is used.

Open EAR-file with WinRAR (You might want to take a backup before your start modifying the EAR-File). Locate the war-file.
The name of the war-file can be different in every project, but there is only one war-file. In out-of-the-box labvantage.ear, the war-file has the name labvantage.war.

Open the war-file by right-clicking, and selecting “View file”.

Open the file web.xml from folder WEB-INF.

a) Locate the following lines:

```
<servlet>
  <servlet-name>WebServices</servlet-name>
  <display-name>WebServices</display-name>
  <servlet-class>org.apache.axis.transport.http.AxisServlet</servlet-
class>
</servlet>
```

b) Add the following lines after the above mentioned lines:

```
<servlet>
  <servlet-name>WSPServlet</servlet-name>
  <display-name>WSPServlet</display-name>
  <description>WSPServlet</description>
  <servlet-class>com.softwarepoint.connector.WSPServlet</servlet-class>
</servlet>
<servlet-mapping>
  <servlet-name>WSPServlet</servlet-name>
  <url-pattern>/wsp</url-pattern>
</servlet-mapping>
```

Save the web.xml, and accept when WinRAR prompts you to save changes. Close the view to the war-file, and accept when WinRAR prompts you to save changes to the archive. Close WinRAR.

Restart the application server.

3 Advanced topics

This chapter includes topics about fine tuning your Connect installation, advanced Connect configuration and troubleshooting. In normal Connect installation this information in seldom needed.

3.1 Connect Application configuration

3.1.1 Setting Connect ports

The ports used by Connect are defined in the file run.properties in the installation folder.

`wspconnect.single.instance.port:`
The port for the Connect client is set to default value 11099, can be changed if needed.

`wspconnect.cajo.port:`
The port for remote calls from, e.g. LabVantage LIMS, the default is 11098. It can be changed if needed.

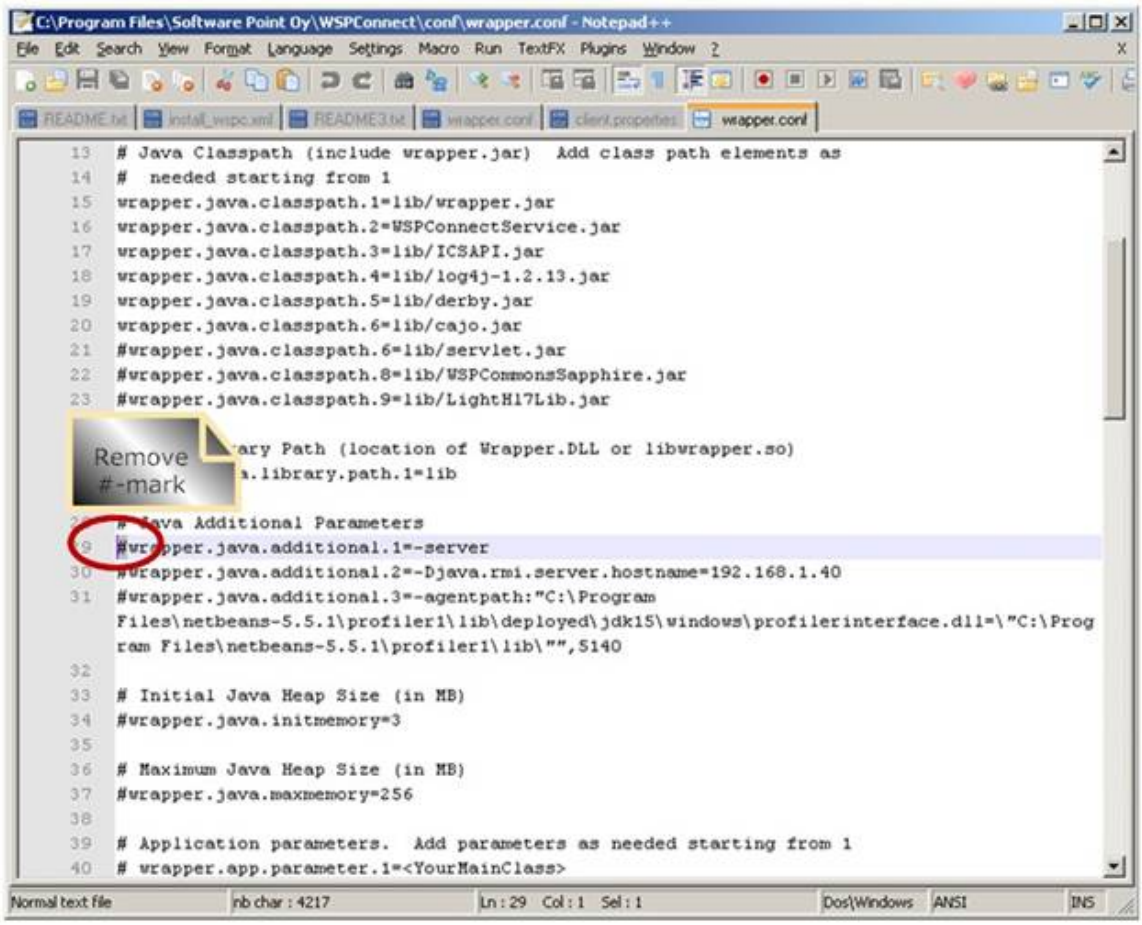
`wspconnect.host:`

Should be set to the same computer name or IP-address running Connect (service or daemon) by the installer. If it's set to localhost it must be changed for remote calls to work.

3.1.2 Server optimization

It is recommended to run java with the `-server` option, if Connect is used under heavy load. To enable this you must use the server JVM, which doesn't come with the JRE installation, but comes with the JDK installation.

To set this option you need to modify the file *wrapper.conf*, which is located in the *conf* folder of the installation directory. This option is commented out by default, you just need to remove the comment mark (#), see below.



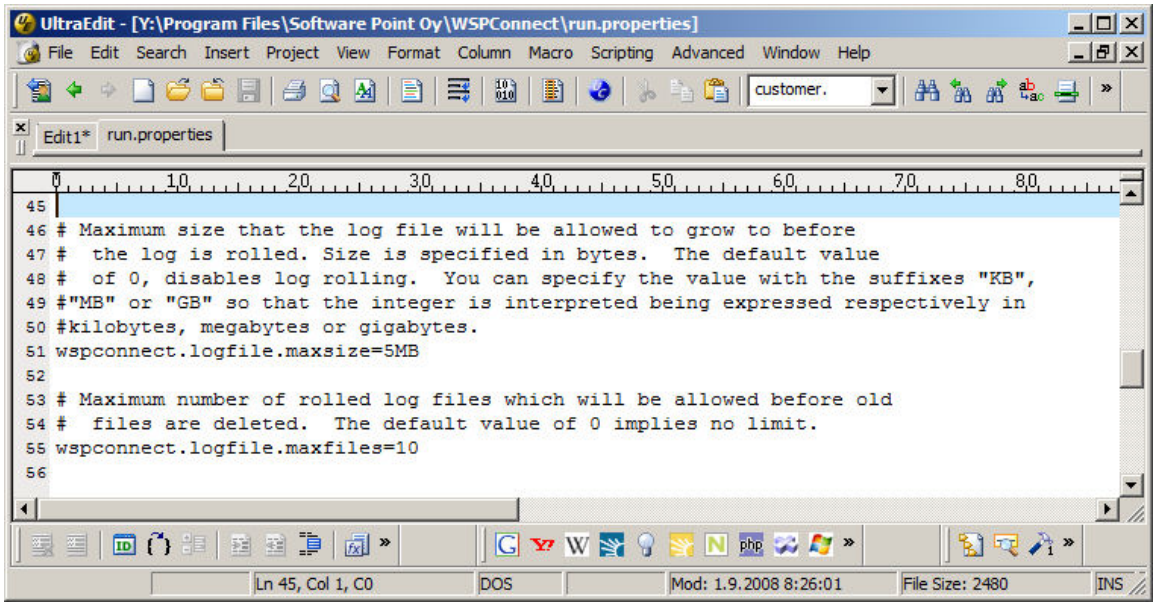
3.1.3 Setting the log file sizes

Connect log files can end up taking too much space, if number of instrument instances is very large. Every instrument instance has its own log file, which will be rolled over to another log file when it becomes full.

For example, when the log file *INSTANCE.log* becomes full, it will be renamed to

INSTANCE.log.1, and the file *INSTANCE.log.1* will be renamed to *INSTANCE.log.2*, until a limit of rolled log files is encountered. After the limit is encountered, the file will not be renamed, it will be deleted.

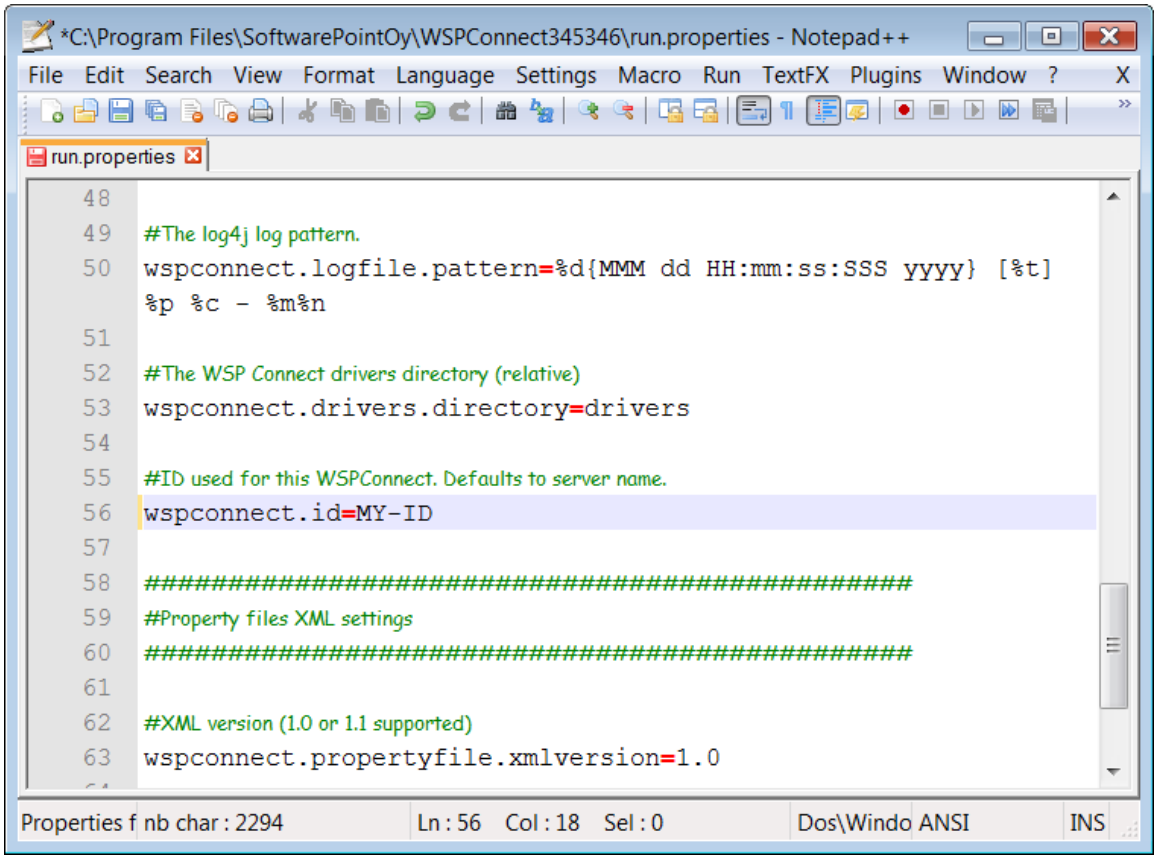
The properties for controlling the log file behaviour are defined in the *run.properties* file shown below.



3.1.4 Setting Connect ID

Connect ID is used to identify the specific Connect on the LIMS side, when Connect is querying LIMS for new tests or worklists.

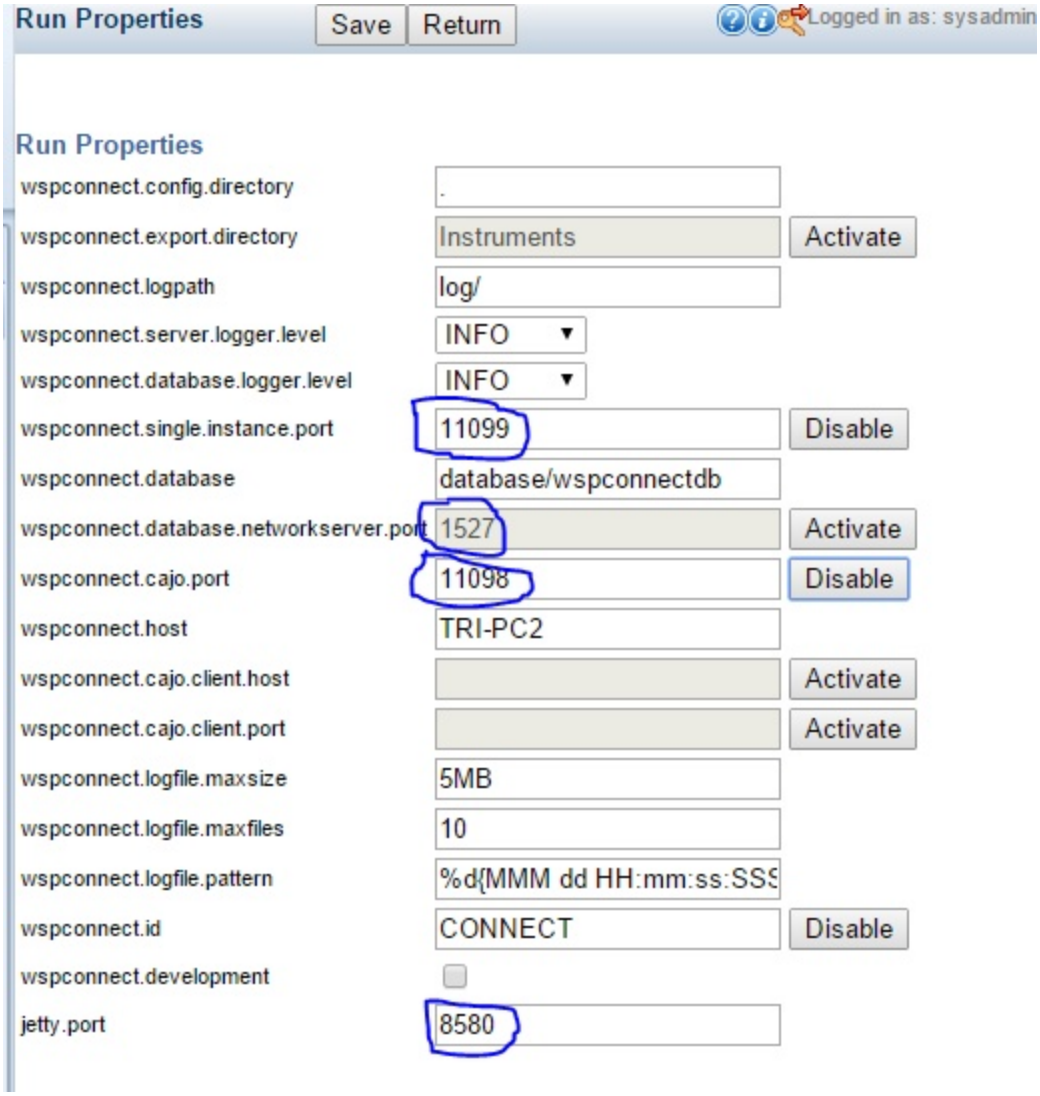
By default, the servername of the Connect installation, in UPPER CASE, is used as the Connect ID. The id can also be specified in the *run.properties* -file. For LabVantage LIMS, the Connect id has to match with a Connect id in the InstrumentServer SDC in the LIMS side.



3.1.5 Multiple Connects on the same Server

Multiple Connects can be installed to the same server, but they must use different ports.

The following is the total list of Connect ports, with their default values:



You can disable wspconnect.single.instance.port and wspconnect.cajo.port, and use HTTP protocol for accessing the Connect client and for Cat I -instruments. This is recommended anyway. (wspconnect.database.networkserver.port is disabled by default.)

So just jetty.port must be different for different Connect Applications.

Also, if Connect is installed as a service in Windows, the services must be named differently. The service name cannot be configured in the installer installer program, so the checkbox "*Install as Service"* should be unchecked when installing additional Connects.

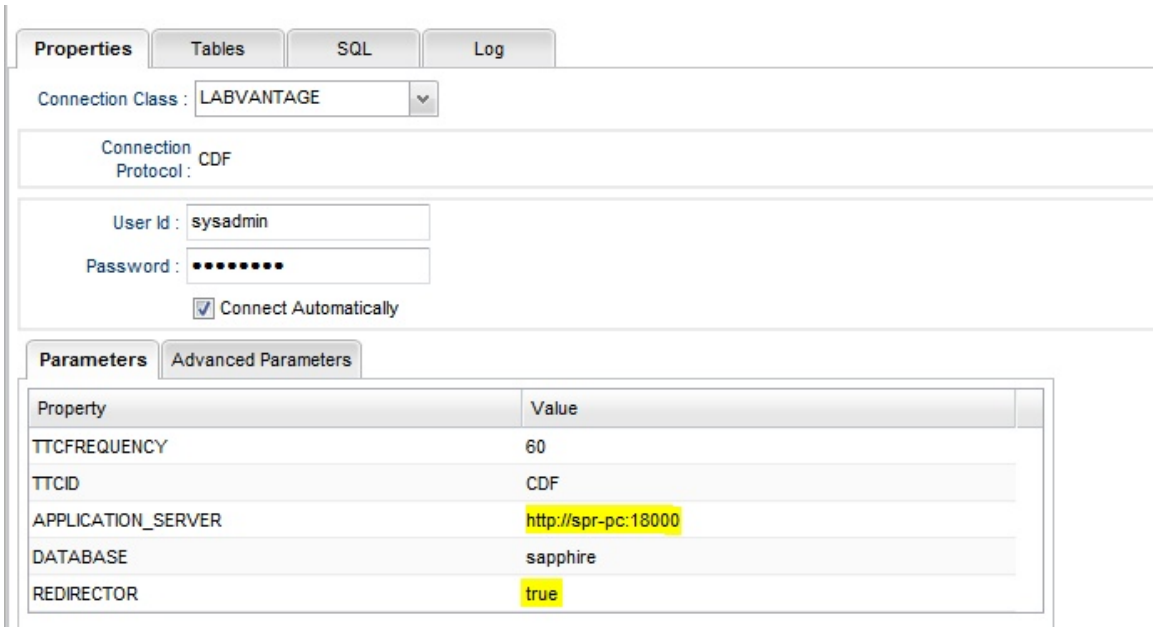
After Connect installer has finished, the service name can be changed in the file *wrapper.conf* located in the *conf*-directory of the Connect installation. The property that defines the name of the service is *wrapper.ntservice.name.* It is set to *WSPConnectService* after installation.

After the name has been configured, the service can be installed by running the bat-file

InstallWSPConnectWrapper-NT.bat, located in the *bin*-directory of the Connect installation.

3.1.6 Using WSPRedirector

It is possible to use WSPRedirector service for load balancing Connect LIMS connections. Also, when this functionality is turner on, connect automatically tries to connect to other configured servers if one that it is currently connected goes down. This functionality requires that WSPRedirector is installed. Once WSPRedirector is installed, functionality can be turned on by setting application_server and redirector properties on Connect client's 'Edit LIMS Connection' page. Application_server property should point to WSPRedirector install server and redirector property value should be 'true'. If redirector property value is other that 'true', redirector functionality is not in use and application_server property value should point to actual application server as usually.



WSPRedirector configuration information including port and server addresses are maintained in httpredirector.properties file located in WSPRedirector\conf folder.

This sort of solution can be used on installations where there are many LV application servers and WSPRedirector is already used for load balancing.

3.1.7 Enable Web Server

Connect comes with an embedded web-server, which is enabled by default. To disable, comment the following line in `run.properties`.

```
#Configuration file for the web server (jetty). Web server will
not be started if the file is  not defined.
wspconnect.webserver.config=jetty.xml
```

Jetty is configured to use port 8580 by default. To change, edit `jetty.xml`.

```
...
        <SystemProperty name="jetty.port" default="8580"/>
...
```

Connect webapplication will enable connecting to instrument drivers (e.g. Category I - drivers) over HTTP, instead of RMI. The servlet is bound to the address with pattern `/connect`, e.g. http://server:8580/connect. The web server can be configured to use HTTPS, if secured connections are needed.

3.1.8 Contents of the installation folder

The folder contains several files defining the properties of Connect, contains implementations of parsers, and contains the files for the temporary database used in Connect. The following list explains the purpose of some of these files and folders.

- **The run.properties file**
  Various properties, such as:
  - Directory where the config files are
  - Log file directory

  ☒Logger level
  ☒Size of the log files
  ☒Port used for RMI-communication
  - Directory for the temporary database

  ☒Time-out for logged users
  ☒Loaded instrument libraries

- **The propertyfile.xml file**
  Defines the instances and temporary tables

  Generally, do NOT edit this file directly! Editing might corrupt the configuration, especially regarding database definition. If replacing the configuration file, you should generally also replace the database.

- **The log folder**
  Contains all logs as text files.

- **The drivers folder**
  Save the implementations of all drivers here.

- **The conf folder**
  The wrapper.conf file, which defines also various other parameters for Java Service Wrapper, e.g. the jar-files loaded, and the java-path used.

- **The wspconnectdb folder**
  Contains the database used by Connect to save temporary tables.

- **The lib folder**

Contains the libraries used by Connect Service.

3.1.9 Uninstalling LVConnect

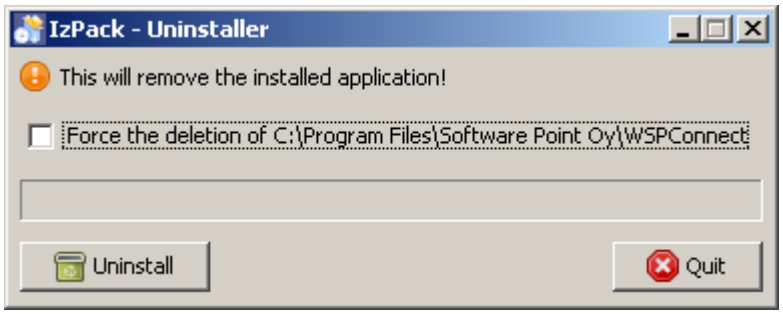
Connect can be uninstalled with the *uninstaller.jar* file in the *Uninstaller* sub folder of the installation directory. Uninstaller will handle stopping and removing the Connect service in Microsoft Windows, but it will not stop Connect that is running in console mode.

WARNING! If you have manually installed the Connect Windows NT Service (i.e. not automatically during the installation), you need to removed it manually with the script "*Uninstall ConnectWrapper-NT.bat*", located in the *bin* subfolder under the installation directory. Remember that you must remove the service before executing the uninstaller.

WARNING! The uninstaller.jar has to be executed with administrator privileges under Windows Vista. To do this, start command prompt as administrator, go to the uninstaller directory, and type *java -jar uninstaller.jar*.

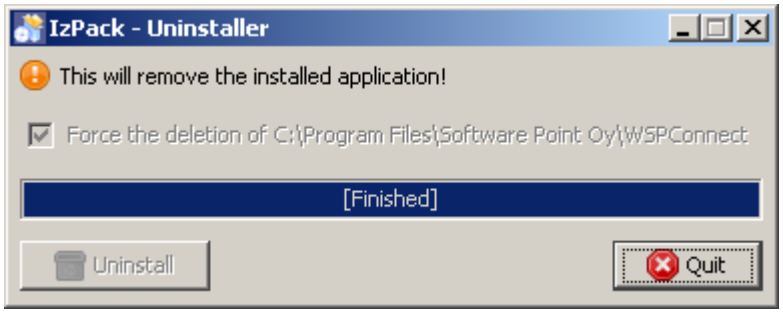
In Unix/Linux, the Uninstaller will not stop Connect, nor remove the script to run Connect as a daemon, all this has to be done manually.

The uninstaller opening window looks like this:



If you check the option "Force the deletion...", uninstaller will remove all the files from the installation folder, i.e. also files that have been modified since the Connect installation.

Click Uninstall to remove the Connect from the computer. Click Quit to finish.



3.2 Clustering Connect

Basically, Connect cannot be clustered. The easiest, default solution is to install LVConnect to one server, and define this server in LIMS. If there are performance concerns, it is better to divide the implemented interfaces to several Connect installations, rather than build one big Connect installation which is clustered in some way.

If you need fault tolerance, this can be achieved by installing more Connect-applications. The way to do this depends on the instrument Category. If the project has both Category 1 and Category 2/3 interfaces, then both solutions must be used.

3.2.1 Category I -interfaces

For Category 1, the connection is made from LIMS to Connect. It is possible to have several Connect-installations so that each node in the cluster has a Connect Application, and the connection from LIMS will be made to localhost. Therefore, each LIMS node will use the Connect Application on the same physical server to make the connection to instruments.

This provides the requested fault tolerance: if a node goes down, the LIMS users will continue to use LIMS on other nodes on the cluster, and they will use the Connect Application in those nodes.

To ease the configuration and maintenance, the Connect Applications should share configuration via a fileshare (propertyfile.xml). If you change any configuration, you need to restart both Connects.

There will be only one Connect-definition in LIMS.

Refer to the image in For Category I instruments to see this setup.

3.2.2 Category II and III -interfaces

For Category II and III, the connection is made from Connect Application to LIMS,and the interfaces are usually continuously connected. Only one Connect should be active at the same time, but there can be other Connect installations which have been configured to use the same configuration. The configuration, and optionally also the Connect temporary database, will be hosted in a shared location.

If the cluster node where Connect is active fails, the interfaces will go down. The connectivity can be restored by starting Connect from some other node in the cluster.

You will only define one Connect in LIMS, and configure all the Connect-installations so that they use the same Connect ID.

Refer to the image in For all instruments to see this setup.

A very robust solution to sharing the Connect configuration and database between the nodes in the cluster is to set up a shared filesystem via GFS.

3.2.3 Summary

So for the most complex configuration where you have both Cat 1 and Cat 2/3 interfaces, you would have

Server 1: Connect (active), Connect-cat-I (active)

Server 2: Connect (off), Connect-cat-I (active)

Fileshare: propertyfile.xml of Connect, propertyfile.xml of Connect-cat-I, possibly the database-folder of Connect.

LIMS: Two Connect-definitions, one for Connect and one for Connect-cat-I. Cat-I is routed to localhost. Port/host/url of the other Connect does not matter.

3.3 Upgrading Connect Installation

Upgrading Connect should be done only if it is absolutely necessary. Only situation that upgrading is recommended is when there is need to use new Connect functionality available in new Connect version that older version of the Connect don't provide. Upgrading Connect can be a quite tricky, because Connect is so tightly coupled with LABVANTAGE LIMS and often Connect is customized in projects. Once Connect interface is installed, it has modified for example XML files that define LABVANTAGE site maps and web pages. There is no reliable way to automatically modify these files with upgrade scrips. So some of the work needs to be done manually.

When upgrading Connect installation, you must make sure to update all the Connect-related components:

- Connect Application
- Connect Interface, which includes (labvantage).ear modifications and Connect exports (Data model, Connect actions, pages, etc.)

New drivers implemented for the project may not exist in the new Connect-version. This may be because the developer has not included the code to the mainline of development (trunk), or just because the driver was developed after the new Connect-version had already been released. If this happens, the driver library may need to be updated after the installation.

Also, if you have modified the LABVANTAGE-Connect interface for the specific installation, you must make sure that the modifications are re-implemented to the new Connect-version. Often the following are modified:

- Connect queries
- Actions in the LIMS-Connect -interface: CIN_InstruResultAction and CIN_ConnectAction

The upgrade from LABVANTAGE 6 to LABVANTAGE 7 is a special case. LVConnect-related jar-files may be removed from the LIMS in the upgrade, and the LABVANTAGEConnect-functionality must be restored by installing the missing items back. The chapter LABVANTAGE 6 to LABVANTAGE 7 Upgrade explains how.

3.3.1 Upgrading Connect Application

Upgrading Connect Application is best to do as a fresh new install.

1. Make a backup of all the existing interfaces configured in Connect. Login to Connect, select all instruments from the checkboxes, and press "Export". Select File System as the option.

2. Check the settings of the LIMS-connection. You need to reconfigure them to the new installation.

3. Check the RunProps -page, if you have modified the run.properties -file (either directly, or using this page), you need to remodify it after installation. This modification is commonly done to for example change the Connect ID.

4. Uninstall, or turn off Connect application.

5. Install new Connect application.

6. Login to Connect. Import a new LIMS Connection, and reconfigure it.

7. Check the RunProps -page and reconfigure it, if needed.

8. Select Import, select File System as the option, and import the XML-export -file you created in step 1.

9. If some of the drivers show up in state Invalid, with Driver Class Invalid, the driver is missing from the new version of Connect. Contact a Connect developer of Software Point support to transfer the driver to the new version. Note that the driver libraries built against older versions of Connect must NOT be used.

3.3.2 Upgrading Connect Interface

3.3.2.1 Upgrading to Connect 7

In Connect version 7 Connect data model has been changed to follow LABVANTAGE module convention. All Connect related components are named with CIN module identifier.
Following process is designed to upgrade Connect interface from Connect 4.x version to newest Connect 7 version.

1. Use LABVANTAGE Console 7 to apply installer CIN_upgrade_to_Connect7_oracle.zip or CIN_upgrade_to_Connect7_sqlserver.zip depending on your database. Installation process is the same as described in chapter Installing the Connect Interface, only the zip package is different.
Installation scripts does the following:
-Alters field names that previous Connect installation has added in LV's SDC's (Instrument, Sample, QcBatch).
-Imports new SDCs (CIN_EdiMessage, CIN_InstruResult, CIN_InstruWorkList, CIN_InstrumentServer, CIN_Query, CIN_SysProperty).

-Copies data from old data model tables to some new cin_ -tables.
-Removes old Connect Tables: w_EdiMessage, w_InstruResult, w_InstruWorkList, w_wspconnect, w_Query, w_SysProperty.
-Removes old Connect web pages.
-Removes old Connect actions, policies, LV-queries, elements and category.
-Installs newest Connect 7 interface version

If installer shows errors, find out the reason before continuing.

2. After running the installer there is two Connect tramlines on Lab Admin site map. Old one is the upper one, the one with "Connect Queries" tram stop. You can remove the old one with LABVANTAGE Web Page Designer. Go to Layout Property Manager (Layout: Generic) and select custom node under the CIN node and remove the old Connect tram line.

3. There is some duplicate fields on Instrument custom -node. Duplicate fields are: Auto validate, Auto release, Handle replicates, Query Mode and Overwrite results.
Remove old Connect fields on Instrument page. New fields are the ones that have a Column id that start with "cin_".
Field Connect Id can also be removed, in Connect 7 it is called Instrument Server Id.

4. Edit LV_InstrumentModel SDC. Open the edit page and remove instrumentmodelmap table from Table(s) tab.

Note that if there is project specific customizations to Connect Interface, there is some more work ahead. Connect Interface parts that are most commonly customized in project include Connect actions and Connect queries.

3.3.2.2 Upgrading other Connect version

Upgrading the Connect interface is done by running the Connect Interface zip package with LABVANTAGE console, in the same way as during installation.

Note that if the project is using customized versions of the Connect actions, pages or queries etc in the LIMS-Connect -interface, the customizations need to be reimplemented to the new version of the LIMS-Connect -interface.

3.3.3 Update Connect Queries

Connect-queries are overwritten in the upgrade. If the Connect-queries have been modified for the installation, as is often the case, the modifications must be re-implemented to the new version of the queries.

To help in the query modifications, the audit is enabled for the queries, and you can see the changes implemented to the queries from the Connect Query list-page. Re-implement your changes to the new queries, and press the Refresh-button in the list-page.

In Connect 7 query id has changed, so there will be two sets of queries in Connect Query

page. Old ones start with wsp and are not in use in Connect 7. After copying possible project specific modifications to new CIN queries, old queries can be deleted.

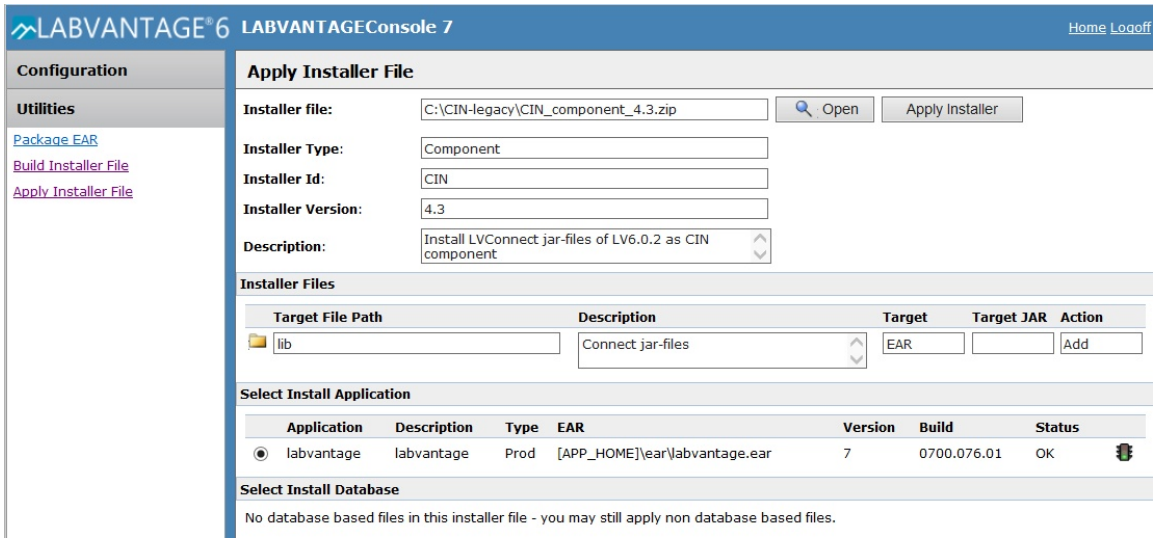
3.3.4 Upgrading troubleshoot

You can find out the version of all the Connect-related components from the Connect Admin -page in LABVANTAGE. Select a Connect-installation, and press "Version Info". If the message shows a version number older than the one where you upgraded, there is likely a problem in the installation.

3.4 LABVANTAGE 6 to LABVANTAGE 7 Upgrade

LABVANTAGE 6.0.2 included the Connect-related jar-files for Connect version 4.3. These jar-files may not be always included in the LABVANTAGE 7. If the customer has implemented some interfaces with Connect 4.3 prior to the LABVANTAGE 7 upgrade, then the upgrade to LABVANTAGE 7 may break them, because the jar-files will be removed from LABVANTAGE.

We provide a small component installation package named CIN_component_4.3.zip, this can be used to install back Connect-related dependencies removed by the LABVANTAGE 7 upgrade. This can be installed with the LABVANTAGEConsole.



Note that upgrading LABVANTAGEConnect should not be attempted without support from LABVANTAGE or Software Point. Upgrading Connect is not required during LABVANTAGE upgrades.

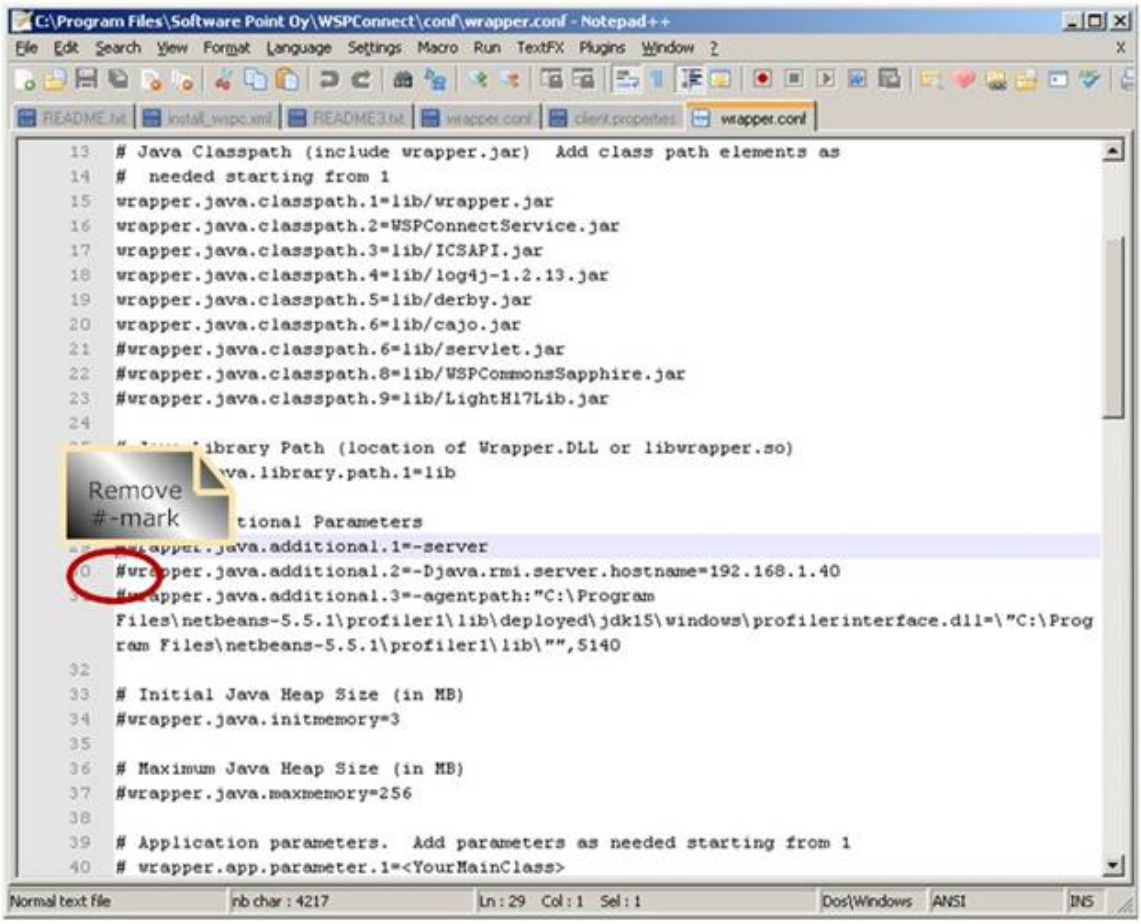
If you choose to go forward with the upgrade to Connect 7, then see the chapter Upgrading to Connect 7 for instructions.

3.5 Troubleshooting

This chapter gives some guidance about troubleshooting the Connect installation.

3.5.1 RMI problems

It is possible, that RMI-connections will not work, if the host incorrectly resolves the host ip-address to localhost, setting it to 127.0.0.1. In this case, you can set give the host ip-address as a parameter to the JVM in conf/wrapper.conf:



The alternative to this is to edit the settings of the host-computer. The following instruction applies to Mandriva Linux, but can probably be applied elsewhere.

If there is the following definition in the *etc/nsswitch.conf* file

```
hosts:      files nis dns
```

change it in a way, that dns is looked up before files:

```
hosts:      dns files nis
```

Or modify *etc/hosts*. The offending line is of the form:

```
127.0.0.1 localhost servername
```

Change it to:

```
127.0.0.1 localhost
192.166.0.123 servername
```

Where the latter IP-address is the actual IP-address of the server.

3.5.2 Server Error Codes

If Connect service / daemon fails to start, it will usually return an error code, which helps to identify the reason for the failure. The meaning of the error codes is defined in the following table:

Error Code:	Reason:	Solution:
5001	The port is reserved. Possibly another application has already running on that port, or Connect is already running.	Check if Connect is already running. If not, change the port used by editing the file run.properties, property single.instance.port.
5003	Failed to establish connection to Connect database. The database may be locked.	Delete the file db.lck in database-folder.
5004	Graceful shutdown of Connect service failed.	In the usual case, just means that e.g. connection from LIMS is not cleaned up. It will still be released by LIMS eventually. Might also be that shutting the Connect process totally fails, and then the process must be killed manually.
5006	The file run.properties could not be found, or is corrupt.	Check the file run.properties, revert to a previous version of the file if possible.
5007	The file propertyfile.xml was corrupted.	Check the file propertyfile.xml for possible problems, revert to a previous version of the file if possible.
5008	Failed to load classes for instrument drivers.	Check that the classes are available under the installation.
5010	Maximum number of configurations allowed by Connect license has been exceeded.	Purchase a new license.

3.5.3 Https

If Connect should connect to LABVANTAGE via https, the certificates have to be provided to it. The exact steps depend on the exact configuration used for https. The following example is for 2-way ssl, using the certificates exported from Sybase EAServer Manager.

The publisher certificate (`.cer` or `.crt`) has to be added to a Java keystore. The

certificate can be added to a java keystore with the *keytool*-program at `%JAVA_HOME%\jre\bin\keytool`, with the following command:

```
keytool -import -alias <server_name> -keystore <keystore> -file <certifica
```

You can use the JVM:s default keystore at `%JAVA_HOME%\jre\lib\security\cacerts` (the default password is "`changeit`"), or create a new keystore.

The server certificate (`.p12`) can be used directly.

The remaining thing is to configure Connects JVM to use these certificates. This is done in the file `conf\wrapper.conf`. (Note that the last two properties are not needed if the JVM:s default keystore is used.) See the following example configuration. (In this the certificates have been copied to the conf-directory.)

```
wrapper.java.additional.3=-Djavax.net.ssl.keyStore=conf/user.p12
wrapper.java.additional.4=-Djavax.net.ssl.keyStorePassword=12345
wrapper.java.additional.5=-Djavax.net.ssl.keyStoreType=pkcs12
wrapper.java.additional.6=-Djavax.net.ssl.trustStore=conf/mykeystore.jks
wrapper.java.additional.7=-Djavax.net.ssl.trustStorePassword=mypasswd
```

After this, restart Connect service.

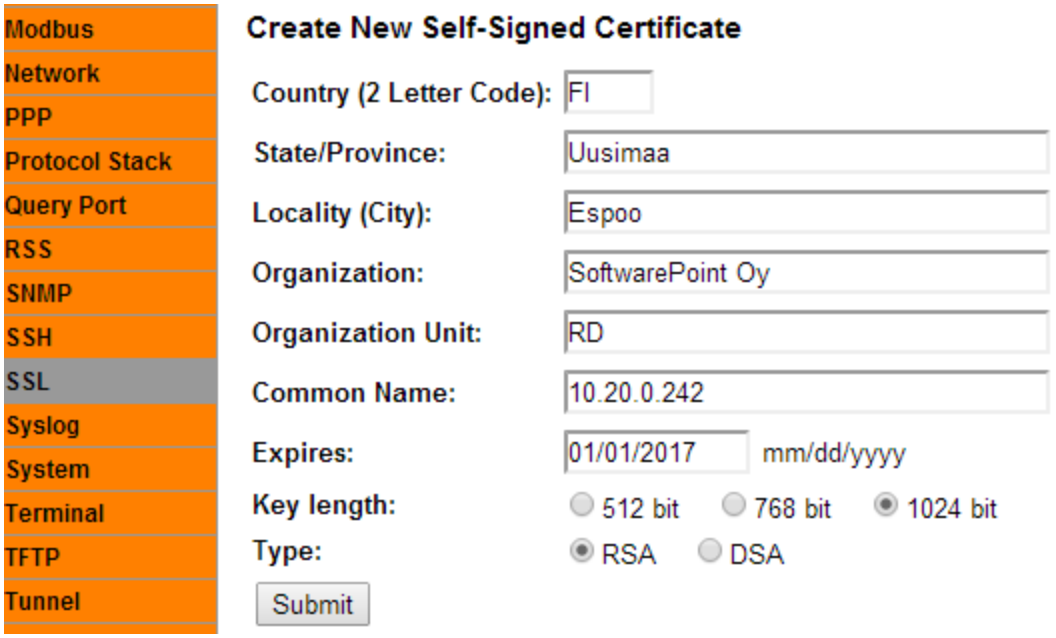
3.5.4 Connect web client doesn't open or acts strangely

If a web browser shows blank page when trying to open Connect web client or some buttons are disabled etc. try clearing the browsing history/cache. This problem has occurred especially with IE browser.

3.5.5 Setting up Lantronix box to use SSL

Open Lantronix Web Manager by entering box's ip-address to browser and give user name/password (default admin/PASS). Choose SSL protocol, Tunnel -> Accept Mode -> Protocol -> SSL.
You can generated certificate by other tools or generate certificate from Web Manager.

Here is how to do it with Web Manager. Open SSL page. Fill in the values, use RSA. Common name should be Server's DNS name or ip-address.
For Example:



Press Submit button. You should see information about newly created certificate. Now the certificate is set to Lantronix box.
Go to XML page, Choose ssl and Export to Browser or to file. Copy the certificate from -----BEGIN CERTIFICATE----- to -----END CERTIFICATE-----. Save certificate to file with .pem extension. Remove all white spaces.

Next is to set certificate to Connect server's java key store. There is instructions how to do that in chapter Https. If you choose to use java default key store, you don't need to modify Connect's wrapper.conf. Just run keytool command, for example:
keytool -import -alias jaguartest -keystore \Apps\Java\jre6\lib\security\cacerts -trustcacerts -file \Test\landronix_cer.pem

To verify that your connection is using SSL handshake you can turn on more logging to wrapper.log by adding following line to wrapper.conf:
wrapper.java.additional.3=-Djavax.net.debug=all:handshake:verbose

Select Communication type "TCP/IP SSL Client" for the driver you are using. Opening the SSL connection can take a few seconds. Setting the driver configuration DISCONNECT_AFTER to false will make calls after the first one faster.

3.5.6 Audit don't work

In SQL Server environments audit tables may no be always automatically created when Connect Interface is installed. If this happens View Audit button won't work on Connect Queries page. To fix this, you can turn Audit Method temporary off and then on again on CIN_Query SDC maint page. This will create Audit table and triggers for cin_query table. Other Connect related table that uses audit by default is cin_instruresult.