



RFID Excellent Applications

The Easy Way to Link RFID Data

User Manual

V 1.7

Validity of the Document

This document covers the version V 1.7 of detego[®] EXPRESS.

Copyright, Trademarks, Disclaimer

Copyright © 2005-2011 RF-iT Solutions GmbH. All rights reserved.

All trademarks or registered marks in this document belong to their respective owners.

Information in this document is subject to change without notice and does not represent a commitment on the part of RF-iT Solutions. This document is provided "as-is state" without warranty of any kind, expressed or implied, including, but not limited to, the particular purpose.

Contact Information

RF-iT Solutions GmbH
Hans-Resel-Gasse 17a, 8020 Graz, Austria
www.rf-it-solutions.com
detego-express@rf-it-solutions.com

Contents

1	INTRODUCTION	5
1.1	detego® EXPRESS.....	5
1.2	System Requirements.....	5
1.3	Use Cases.....	6
2	DETEGO® EXPRESS CONNECTIONS	7
2.1	Device Control.....	7
2.2	Interfaces to IT - Backend Systems.....	7
3	DETEGO® EXPRESS GUI APPLICATION	9
3.1	Download and Installation.....	9
3.2	Start.....	10
3.3	GUI – Graphical Interface.....	10
3.4	Exit.....	12
4	LICENSE	13
4.1	Request License.....	13
4.2	Activate License.....	13
5	SUPPORT	14
5.1	Request Support.....	14
5.2	Feedback.....	14
5.3	About Information.....	14
6	DETEGO® EXPRESS CONFIGURATION TOOL	15
6.1	Parameter Presentation.....	15
6.2	Parameter Description.....	16
6.2.1	General Parameter.....	16
6.2.2	Connection Parameter.....	17
6.2.3	Filter.....	17
6.2.4	Logging.....	18
7	DETAILED INTERFACE DESCRIPTIONS	19
7.1	File.....	19
7.2	Database.....	20
7.2.1	OLE DB.....	20
7.2.2	ODBC.....	20
7.2.3	MySQL.....	21
7.2.4	Microsoft SQL Server.....	22
7.2.5	ORACLE.....	23
7.3	Web Service.....	23
7.3.1	Polling Mode.....	24
7.3.2	Event Triggered Mode.....	25

7.4	ALE Web Service	26
7.5	HTTP Request.....	29
7.6	Microsoft Message Queuing (MSMQ).....	30
7.7	DA OPC Connector	31
7.8	Process	32
8	THREE ADDITIONAL WAYS TO START DETEGO® EXPRESS	33
8.1	Log Viewer	33
8.2	Console Application.....	34
8.3	Windows Service.....	35
9	TOOLS	36
9.1	You-R® OPEN Verification Client	36
9.1.1	Presentation	38
9.1.2	Functionality	38
9.2	EPC GEN2 Application.....	40
10	ENHANCED WEB SERVICE INTERFACE	42
10.1	EPC Reader Web Service.....	42
10.2	Facade .Net.....	43
11	GLOSSARY	44
12	FIGURES	46
13	TABLES	47

1 Introduction

1.1 detego® EXPRESS

detego® EXPRESS is a cost effective and professional way to transfer accurate data from a number of RFID devices to the IT-Backend System.



Figure 1-1: detego® EXPRESS Architecture

Features:

- **Large Amount of Devices**
All major RFID devices are available
- **Different Technologies of Devices**
UHF, HF, LF and barcode
- **Variety of Interfaces**
Database interfaces, file output, web service interfaces
- **Transaction Security**
Data is buffered when connection is not established.
- **Defined Data is provided for IT-Backend System**
ID, timestamp and location
- **Filtering of Data**
Distinct and EPC pattern filter
- **Customizing**
Additional configuration software provides the possibility to customize the solution.
- **Enhanced EPC Web Service Interface**
A set of specific functions is available out of the box, e.g. activation or deactivation of the antenna field.
- **Additional Tools**
Get additional information or functionality by using included tools.
- **Four Week free Evaluation License**

1.2 System Requirements

- Computer** Only a few system requirements are necessary to start **detego® EXPRESS**:
- Microsoft Windows XP SP2 or higher
 - .Net Framework 2.0 or higher
- Reader** No physically reader is necessary when testing with a simulator version.
Otherwise one of the [current supported devices](#) must be available.

1.3 Use Cases

detego[®] EXPRESS links Auto-ID data to any backend system. It covers a range of use cases:

- Identification points along a production or assembly line
- Packing table solutions
- Returnable items in closed loop applications
- Inbound or outbound solutions
- Portal gates
- ...

2 detego® EXPRESS Connections

In the following the connections to the device as well as to the IT – Backend are described shortly.



Figure 2-1: detego® EXPRESS Connections

2.1 Device Control

A variety of different devices supporting LF, HF, UHF, barcode and handheld modules are available with **detego® EXPRESS**. If you plan to use a device that is currently not included, do not hesitate to inform us and send a request to detego-express@rf-it-solutions.com.

More detailed information (in form of device related information sheets) about selectable devices is available.

A complete list of all currently available devices for **detego® EXPRESS** you will find at RF-iT Solutions homepage:

<http://www.rf-it-solutions.com/en/produkte/detego-express/devices.html>

2.2 Interfaces to IT - Backend Systems

Interfaces

In order to transfer aggregated data to the IT-Backend system **detego® EXPRESS** lets you choose between a number of specific interfaces:

- File
- Databases
 - OLE DB
 - ODBC
 - MySQL
 - Microsoft SQL Server
 - ORACLE
- Web Service
- ALE Web Service
- HTTP Request
- OPC Connector
- Process

Additional interfaces such as ORACLE, MySQL, ODBC, message queuing or TCP/IP Sockets are available on request (detego-express@rf-it-solutions.com).



Note:

When using handhelds only the following interfaces are available: file, HTTP request and MS SQL Webservice.

Data

Per default, the following pre-defined set of data is sent to the IT backend system:

- ID
- Timestamp
- Location (user specified name of the used RFID device)



If there is need to transfer additional data to the backend system ask for our advanced development platform product [detego® You R OPEN](#).
(detego-express@rf-it-solutions.com)

You will find detailed information about interfaces in chapter 7 Detailed Interface Descriptions.

3 detego® EXPRESS GUI Application

detego® EXPRESS provides different device and interface dependent applications that announce the current state of your system architecture and also depict the data that is sent to the backend system at the same time.

You-R® OPEN Tube Builder

All these applications are created with help of the You-R® OPEN Tube Builder, part of the development platform detego® You-R® OPEN. Simply put, the You-R® OPEN Tube Builder provides system integrators or developers the possibility to create end-customer specific applications by oneself.



Send a request to RF-iT Solutions to get more information concerning our advanced development platform product [detego® You R OPEN](http://www.rf-it-solutions.com/en/produkte/detego-express.html).
(detego-express@rf-it-solutions.com)

3.1 Download and Installation

Download

Download **detego® EXPRESS** as a zipped file directly from the following link:

<http://www.rf-it-solutions.com/en/produkte/detego-express.html>

All you have to do is to register yourself with name and email address.

Installation

After extracting the files you will find the following file structure on your computer:

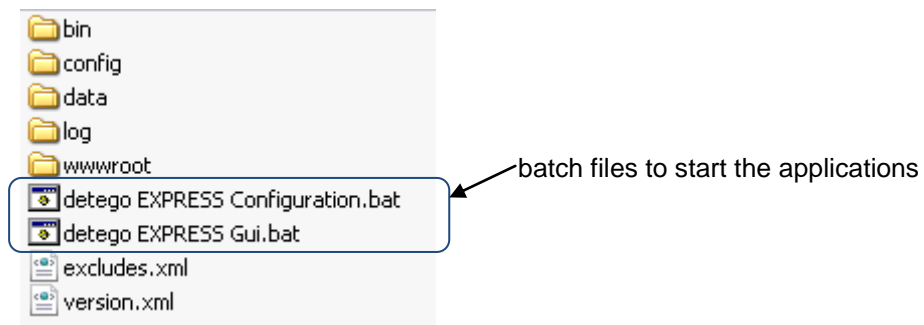


Figure 3-1: detego® EXPRESS File Structure

Warning:



Windows 7,
Vista

When using **Windows7** or **Vista**, administrator rights are mandatory to run the batch files.

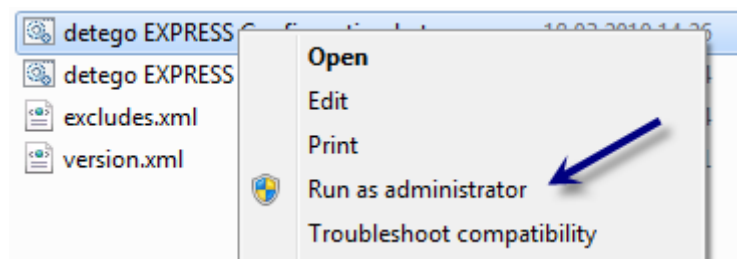


Figure 3-2: Windows7 - Run as administrator

If the execution of the batch files fails, start the program directly using administrator rights.

GUI: \bin\Tube.DetegoExpress.Gui.exe
Configuration Tool: \bin\Tube.DetegoExpress.GuiConfig.exe

3.2 Start

Before starting **detego® EXPRESS** GUI application some connection parameters have to be set first.

detego EXPRESS Configuration.bat Double click the file **detego EXPRESS Configuration.bat** to open the **detego® EXPRESS** Configuration application. See chapter 6 to learn how to change parameters.

Change the connection parameter to your device as described in chapter 6.2.2. Depending on your selected interface type to the IT – Backend System change the corresponding parameters as described in chapter 7.

After having set the connection parameters the application can be started:

detego EXPRESS Gui.bat Double click the file **detego EXPRESS Gui.bat** to open the **detego® EXPRESS** graphical interface application.

Tags placed inside the antenna field of the device are automatically displayed within the application (with the exception of simulated devices, where virtual tags are displayed).

3.3 GUI – Graphical Interface

The following figure shows an example of the graphical interface.

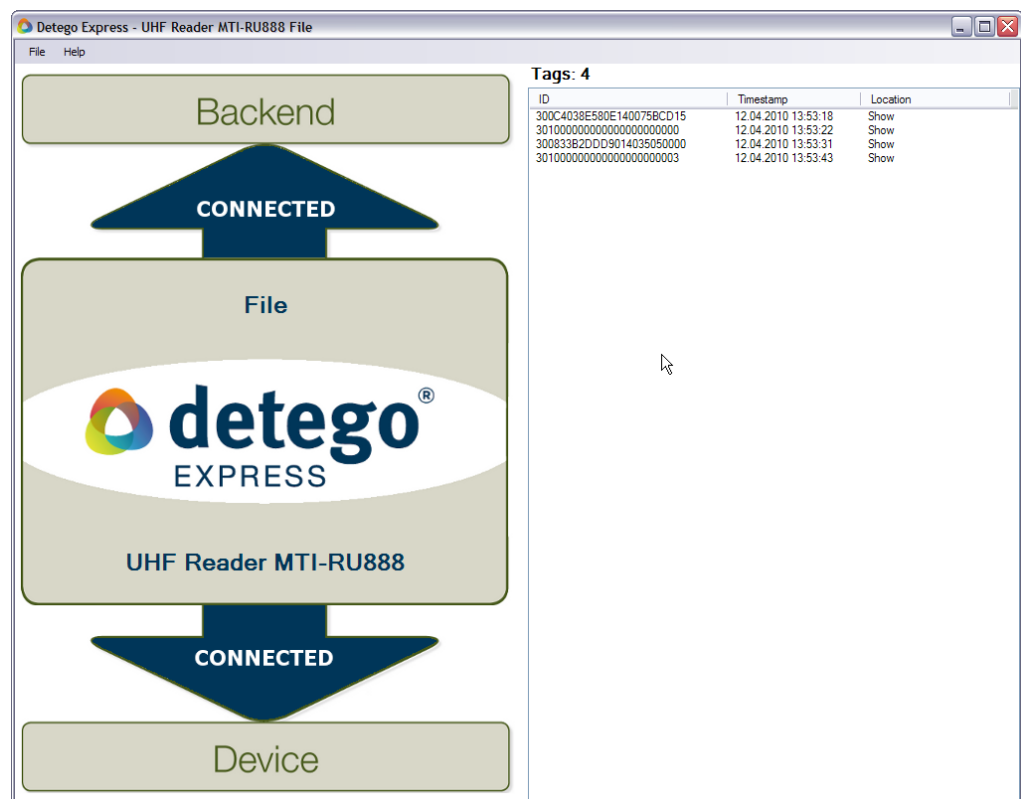


Figure 3-3: Running detego® EXPRESS Application

The left side of the application gives you a simple glimpse of the application architecture. An RFID device transfers data to a backend system using **detego® EXPRESS**. Which components are involved is displayed in the graphic. Within the above example an "UHFReader MTI-RU888" (RFID ME™) is used as device and the interface "File" writes the data into a file.

Connection

It is important to know if the device as well as the selected backend system is connected. This is visualized within the application.

Connected

Blue filled arrows with the text "CONNECTED" signalize that the system works well.

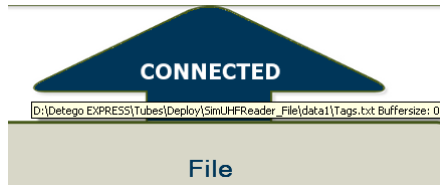


Figure 3-4: Connected Information

When moving the cursor over the arrow connection information is displayed. The last block always contains the buffer size. "Buffer size: 0" means that all IDs are transferred to the selected backend system. The example above displays the filename where the data is written, when the interface "File" is used.

Not Connected

If no connection can be established the arrow remains unfilled and the text "NOT CONNECTED" signalizes the connection state of the device or interface.

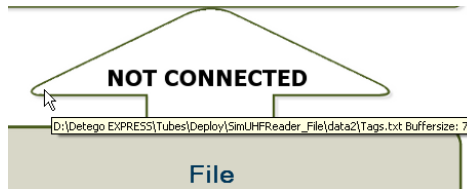


Figure 3-5: Device - Not Connected

When moving the cursor over the arrow connection error information is displayed. All read data is stored within the offline buffer and will be transferred when the connection has reestablished. The data will remain in buffer even when the application is closed in between.

Log Viewer – Error case

If more connection error information is required, start the application with the provided tool "Log Viewer" as described in chapter 8.1 or have a look to the logging files in the subfolder "log".

Tags

The right side of the application lists all tag IDs currently placed in the antenna field of the RFID device.

Tags: 4

ID	Timestamp	Location
3006FB63AC1B3941EC880176	24.03.2010 08:48:47	Readpoint 1
3005FB63AC1F3841EC865033	24.03.2010 08:48:48	Readpoint 1
3005FB63AC1F3841EC880467	24.03.2010 08:59:29	Readpoint 1
3005FB63AC1B3941EC880078	24.03.2010 09:01:14	Readpoint 1

Figure 3-6: Tag List

Provided information is:

Column	Description
ID	EPC (Electronical Product Code) written on the tag
Timestamp	Data and time when the EPC is detected
Location	Name of the device that detected the tag

Table 3-1: Tag Information

3.4 Exit

To close the application select **File|Exit** .

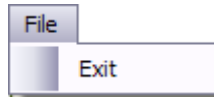


Figure 3-7: File Menu

The other way is simply to close the window.

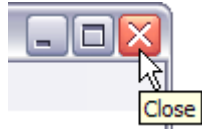


Figure 3-8: Close Window (Application)

4 License

When starting **detego**[®] **EXPRESS** application for the first time, the evaluation license period - valid for four weeks after first run of the application - will start.

Request a **single workplace license** to use this application without any constraints.

4.1 Request License

There are two ways to get an unlimited license.

1. At the homepage: "Request License"
(<http://www.rf-it-solutions.com/en/produkte/detego-express.html>)
2. Within the **detego**[®] **EXPRESS** application select the **Help** menu.

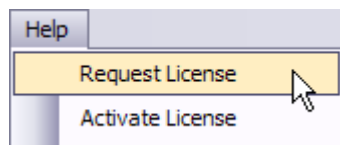


Figure 4-1: Help Menu - License

Select **Help|Request License** within your **detego**[®] **EXPRESS** application.

Request Unlimited License

Request an unlimited license for a single workplace by clicking [Request License](#) within the **Help** menu or use the provided hyperlink here. Fill in the displayed form and request the license.

license.yol

After paying the license fee you receive the license file named **license.yol** per email.

4.2 Activate License

Select **Help|Activate License** within your **detego**[®] **EXPRESS** application.

license.yol

In the file open dialog browse to your **license.yol** file.



Figure 4-2: Open license.yol

Select the license file and the license will be installed automatically. After the next restart of the **detego**[®] **EXPRESS** application the unlimited license is activated.

5 Support

5.1 Request Support

If the application does not work as expected, do not hesitate to contact support. Please note, that **detego[®] EXPRESS** includes a free support request! Further requests are charged by effort according to our current price list.

Sending Request To send a request for support click [Request Support](#) within the **Help** menu or use the provided link therein. An email client will open a new mail to our support with a specific subject.

5.2 Feedback

We are looking forward to receiving your feedback. Please take your time to inform us about your needs. This will help us to further improve our products according to your requirements.

Sending Feedback To send a feedback message click [Feedback](#) within the **Help** menu or use the provided link here. An email client will open a new mail to our support with a specific subject.

5.3 About Information



Figure 5-1: About Dialog

License Limitation When using a trial license, the end of the validation period is visualized. After purchasing an unlimited license, this fact will be displayed here.

Version Number The version number is useful for any support request.

Homepage A direct link to the RF-iT Solutions homepage (<http://www.rf-it-solutions.com>) is also placed within this dialog.

6 detego® EXPRESS Configuration Tool

A number of device specific parameters can be adjusted according to your needs.

The **detego® EXPRESS** Configuration application provides a comfortable tool for editing configuration parameters.

Start
detego EXPRESS Configuration.bat

Double click the file **detego EXPRESS Configuration.bat** to open the **detego® EXPRESS** Configuration application.

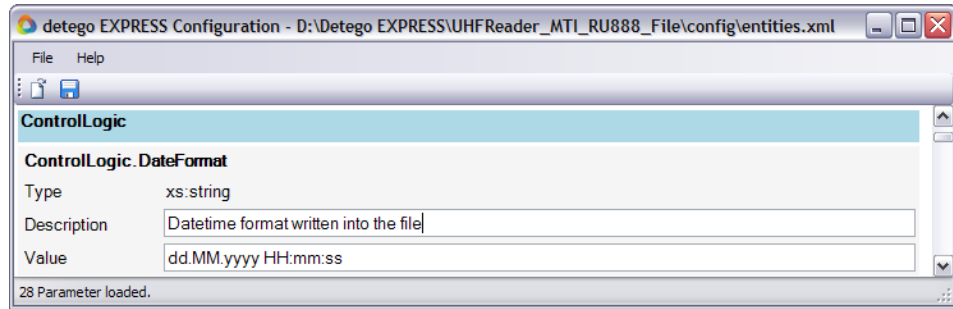


Figure 6-1: detego® EXPRESS Configuration Tool – Example

6.1 Parameter Presentation

All adjustable parameters are listed grouped by logical units. Groups as well as parameters are displayed in alphabetical order.

Parameter Group Groups are displayed with blue background, such as shown in the figure below.

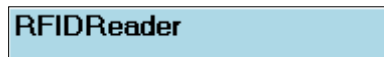


Figure 6-2: Parameter Group Name

Parameter Each parameter is represented with its type, description and value.

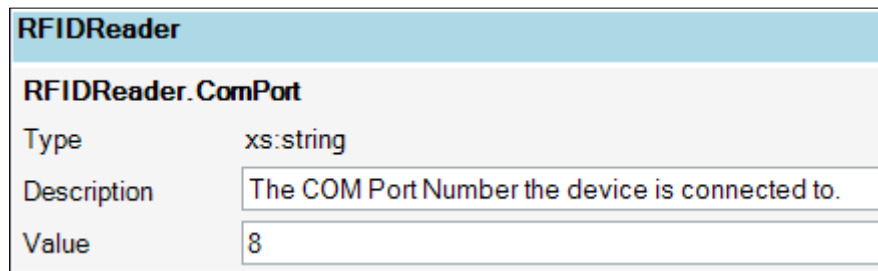


Figure 6-3: Parameter Representation

Type The type of the parameter is displayed as it is used within the code.

Type	Description
xs:string	Text value
xs:int	Integer value
xs:boolean	Boolean value: True or False (unchecked checkbox means False)

Table 6-1: Parameter Types

Description The description gives you a short glimpse into the usage of the parameter.

Value Change the parameter value by typing the new value into the field or check/uncheck the checkbox for Boolean values.



Figure 6-4: Changing Boolean Variables (unchecked box means false)

Save To save the changed values click the Save icon or select File|Save from the menu.

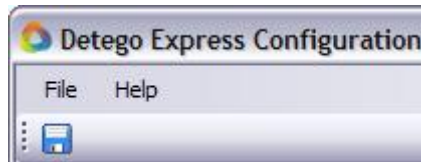


Figure 6-5: Save Configuration Values

6.2 Parameter Description



Note: The following chapters describe all parameters commonly used. The tables list the parameter name in the following structure:

[group name.parameter name]

Example: **RfidReader.Active**

6.2.1 General Parameter

These parameters are provided for all devices. Please note, that these parameters may be listed within different groups.

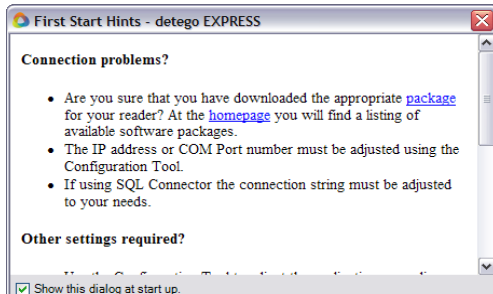
	Parameter	Description
Accept Unsupported Version	RFIDReader.AcceptUnsupportedVersion	Define if the device is started even if an unsupported version is detected. This option is selected per default. In case of errors it may be that the used firmware version is not supported. Check within the log file the version and compare with the versions listed in the device specific documentation.
Tag Alive Time	RFIDReader.TagAliveTime	Define the minimum time in milliseconds a tag is held in the internal tag list.
Location	ControlLogic.Location	Define the name of the location where tags are reported.
Port	Component.Port	Define the port of detego® EXPRESS webservice.
Show Tip	DetegoExpress.Showtip	Define if the start is shown or not. When unselecting the checkbox at start time of the application this parameter is affected. 

Table 6-2: General Parameter

6.2.2 Connection Parameter

Depending on the device different connection types are supported. Therefore different parameters have to be adjusted.

6.2.2.1 Serial Connection

Using a serial connection the following parameters have to be adjusted:

	Parameter	Description
COM Port	RFIDReader.ComPort	Define the COM Port number your device is connected to.
Baud Rate	RFIDReader.BaudRate	Define the baud rate the driver uses for device communication.

Table 6-3: Serial Connection Parameter

6.2.2.2 TCP Connection

For TCP connection the following parameters must be adjusted:

	Parameter	Description
Host Address	RFIDReader.HostAddress	Define the name or IP address of the host the device is connected to.
Port	RFIDReader.Port	Define the Port number your device is listening on.

Table 6-4: TCP Connection Parameter

6.2.3 Filter

detego® EXPRESS enables you to filter specific data. Per default all filters are deactivated.

6.2.3.1 Distinct Filter

A good example for demonstrating the usefulness of the distinct filter is any time tracker solution. After recognizing a tag ID for the first time all subsequent detections of the tag are filtered for the duration of one minute.

Two filter modes are provided:

- Time

Example

A time based distinct filter can be used for a time tracker solution. After recognizing a tag ID, all subsequent reads are filtered for the specified time.

- NumberOfTags

Example

When scanning a box of clothes each article should be recognized only once. To control memory usage, define a threshold number of maximum seen different tags.

To activate the distinct filter and to define the filter criteria use the following parameters:

	Parameter	Description
Active	RFIDReader.DistinctFilterActive	Mark the check box for activating the filter.
Filter Mode	RFIDReader.DistinctFilterMode	Switch mode between time window (Time) and number of different tags stored in buffer (NumberOfTags).
Number of Tags	RFIDReader.DistinctFilterNumberOfTags	Define the number of last seen different tags. When scanning more than the defined number of

	Parameter	Description
		different tags the first tag will be dropped out of the list.
Timeout	RFIDReader.DistinctFilter Timeout	Define the time in ms when the tag with the same id is seen the next time.

Table 6-5: Distinct Filter Parameter

6.2.3.2 EPC Pattern Filter

The EPC Pattern Filter offers the possibility to filter tags with specific EPC patterns. For detailed information about EPC pattern see the EPCglobal Tag Data Standards (<http://www.epcglobalinc.org/standards/tds>).

Example

When scanning a box with different articles only one article should be scanned and counted. Define an EPC pattern to filter unwanted tags and to fulfill this task.

	Parameter	Description
Active	RFIDReader.EpcPatternFilterActive	Mark the check box to True for activating the filter.
Pattern String	RFIDReader.EpcPatternFilterPatternString	Define the filter string according to the standards.

Table 6-6: EPC Pattern Filter Parameter

6.2.4 Logging

In the case of an error, log files are very helpful. The following parameters determine how much information is logged to the log file.

Log Levels

All logging parameters are grouped in section **Tube**.

The following log level values can be set: ERROR, WARN, INFO, DEBUG. Please note that the level DEBUG generates the most output, whereas the level ERROR only logs error notifications to the log file.



Note: In the case of an error change all available log levels to DEBUG, then run the application again and attach the packed log file directory to your **support** request.

7 Detailed Interface Descriptions



Note: The following chapters provide an overview of currently implemented interfaces. Interface specific parameters are described in the following chapters. Detailed information on how to adjust these parameters are available in chapter 7 of this document.

7.1 File

The simplest way to process data is to write them to a file.

Default file name "data\Tags.txt"
 The file path and name must be defined with help of the **detego[®] EXPRESS** Configuration tool (see chapter 6 **detego[®] EXPRESS** Configuration Tool). By default the file `Tags.txt` is stored inside the subfolder `data` within the installation folder. For more information see parameter description in the Table 7-2: File Interface Parameter.

Mobile device
 When using a mobile device there are differences in handling. Please take note of the device specific documentation.

Offline Buffer
 All captured data is stored in an offline buffer as long as no connection to the file can be established. E.g. you are writing to a network drive that is not available at the moment.

File Format
 Each entry inside the destination file is structured as follows:

Column	Description
ID	EPC of the tag
Timestamp	Date and time when the tag is seen at the RFID device. The format is defined by the ControlLogic.DateFormat configuration parameter.
Location	user specified name of the used RFID device (ControlLogic.Location)

Table 7-1: File Structure

Example
 If using comma as delimiter character a possible output entry could look like this:

```
2F0384233443442003D25A69,11.03.2010 10:28,Readpoint 1
```

Parameter
 The following table describes all file specific parameters.

	Parameter	Description
File Path	ControlLogic.FilePath	Define either the absolute or relative path to the destination file. Do not insert the leading slash when using a relative path. Make sure that the path you inserted exists. E.g.: "data\Tags.txt"
Delimiter	ControlLogic.Delimiter	Define the delimiter character for the output file.
Date/Time Format	ControlLogic.DateFormat	Define the date and time format string used for the file output.
Offline Buffer Retry	ControlLogic.OfflineBufferRetry	Define the time in milliseconds when the offline buffer will retry to transmit data.

Table 7-2: File Interface Parameter

7.2 Database

7.2.1 OLE DB

OLE DB (Object Linking and Embedding, Database) separates the data store from the application that needs access to it through a set of abstractions that include the data source, session, command and row sets. This was done because different applications need access to different types and sources of data and do not necessarily want to know how to access functionality with technology-specific methods. (http://en.wikipedia.org/wiki/OLE_DB)

Addon

For testing purpose an Access database named **DB.accdb** is provided. To use it change the database connection string to this database.

Define the connection string and the insert statement and parameter according the database type to insert data into database.

Parameter

The following table describes all OLE DB specific parameters.

	Parameter	Description
Connection String	DBConnectorBackend.ConnectionString	Define the connection string to connect to the database. e.g.: <code>Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\detego Express\Tubes\SIMReader_DB\SIMReader_DB\AddOn\DB.accdb;Persist Security Info=False;</code> Connection strings can be found under http://www.connectionstrings.com .
EPC Parameter	ControlLogic.EPCParameterName	Define the name used for the EPC within the insert statement. E.g.: <code>@EPC</code>
Location Parameter	ControlLogic.LocationParameterName	Define the location name used for the device within the insert statement. E.g.: <code>@Location</code>
Read Date Parameter	ControlLogic.ReadDateParameterName	Define the name used for the date when the tag is read within the insert statement. E.g.: <code>@ReadDate</code>
SQL Statement	InsertID.Query	Define the insert statement that is valid for your used database. The following example shows a valid insert statement using a MS ACCESS database. The parameters used must be written in the same way as listed in the parameters above. E.g.: <pre>INSERT INTO Tags ([EPC] , [ReadDate] , [Location]) VALUES (@EPC , @ReadDate , @Location)</pre>

Table 7-3: OLE DB Interface Parameter

7.2.2 ODBC

The Microsoft Open Database Connectivity (ODBC) interface is a C programming language interface that makes it possible for applications to access data from a variety of database management systems (DBMSs). ODBC is a low-level, high-performance interface that is designed specifically for relational data stores.

Create your data source using **Control Panel|Administrative Tools|Data Sources (ODBC)**. In our example it is named **detegoEXPRESS** and connects to an SQL Server.

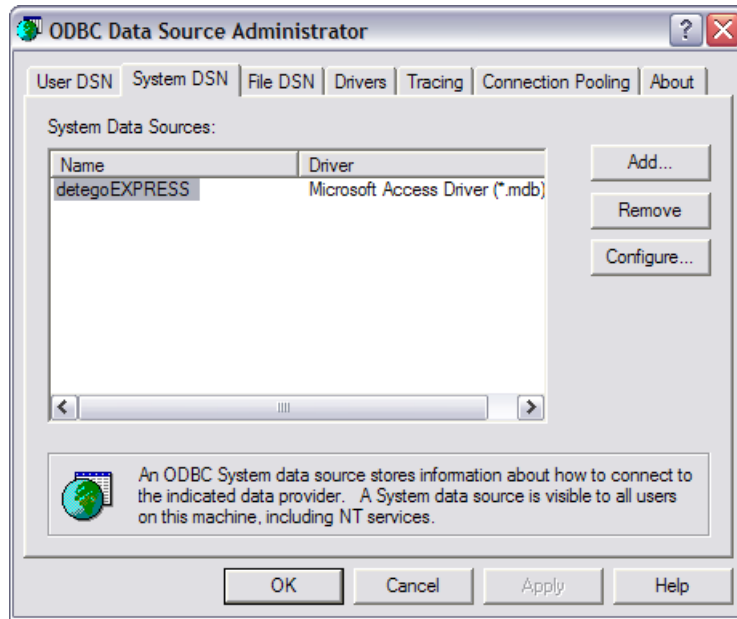


Figure 7-1: ODBC Data Source Definition

Addon
Example Database

Within the Addon folder an example database is provided: ... \Addon\DB.mdb

Configure your data source e.g. using this MS ACCESS database. If you do it in this way no parameter must be changed for running **detego® EXPRESS** application, otherwise define the connection string and the insert statement and parameter according the database type to insert data into database.

Parameter

The following table describes all ODBC specific parameters.

	Parameter	Description
Connection String	DBConnectorBackend.ConnectionString	Define the connection string to connect to the database. e.g. DSN=detegoEXPRESS
EPC Parameter	ControlLogic.EPCParameterName	Define the name used for the EPC within the insert statement. E.g.: EPC
Location Parameter	ControlLogic.LocationParameterName	Define the location name used for the device within the insert statement. E.g.: Location
Read Date Parameter	ControlLogic.ReadDateParameterName	Define the name used for the date when the tag is read within the insert statement. E.g.: ReadDate
SQL Statement	InsertID.Query	Define the insert statement that is valid for your used database. The following example shows a valid insert statement using a MS Access database. The parameters used must be written in the same way as listed in the parameters above. E.g.: INSERT INTO Tags (EPC ,ReadDate ,Location) VALUES (? , ? , ?)

Table 7-4: ODBC Interface Parameter

7.2.3 MySQL

MySQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases.

Define the connection string and the insert query to insert data into the database.

Addon For testing purpose an SQL script named **DB.sql** will create a test database within your installed MySQL database.

Parameter The following table describes all MySQL specific parameters.

	Parameter	Description
Connection String	DBConnectorBackend.ConnectionString	Define the connection string to connect to the database. e.g.: Server=192.168.50.142;Database=detegoEXPRESS_mysql;Uid=root;Pwd=root
EPC Parameter	ControlLogic.EPCParameterName	Define the name used for the EPC within the insert statement. E.g.: ?EPC
Location Parameter	ControlLogic.LocationParameterName	Define the location name used for the device within the insert statement. E.g.: ?Location
Read Date Parameter	ControlLogic.ReadDateParameterName	Define the name used for the date when the tag is read within the insert statement. E.g.: ?ReadDate
SQL Statement	InsertID.Query	Define the insert statement that is valid for your used database. The following example shows a valid insert statement using a MySQL database. The parameters used must be written in the same way as listed in the parameters above. E.g.: <pre>INSERT INTO Tags (EPC , `Date` , Location) VALUES (?EPC ,?Date ,?Location)</pre>

Table 7-5: MySQL Interface Parameter

7.2.4 Microsoft SQL Server

Microsoft SQL Server is a relational model database server produced by Microsoft. An SQL Connector defines a connection to this database.

Define the connection string and the insert query to insert data into the database.

Addon For testing purpose an SQL script named **DB.sql** will create a test database within your installed Microsoft SQL Server Express Edition.

Parameter The following table describes all Microsoft SQL Server specific parameters.

	Parameter	Description
Connection String	SQLConnectorBackend.ConnectionString	Define the connection string to connect to the database. e.g.: initial catalog=DetegoEXPRESS_MSSQL;Persist Security Info=False;Integrated Security=SSPI;server=localhost;Connect Timeout=3
SQL Statement	InsertID.Query	Define a valid insert query for your database. The following example shows an insert statement using a MS ACCESS database. The parameters used must be written in the same way as listed in the parameters above. E.g.: <pre>INSERT INTO [SIM_MSSQL].[dbo].[Tags] ([EPC] , [Date] , [Location]) VALUES (@EPC ,@Date ,@Location)</pre>

Table 7-6: MS SQL Server Interface Parameter

7.2.5 ORACLE

The **ORACLE Database** is a relational database management system (RDBMS) produced and marketed by Oracle Corporation.

Define the connection string and the insert query to insert data into the database.

Addon For testing purpose an SQL script named **DB.sql** will create a test database within your installed ORACLE database.

Parameter The following table describes all ORACLE specific parameters.

	Parameter	Description
Connection String	DBConnectorBackend.ConnectionString	Define the connection string to connect to the database. e.g.: Data Source=(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=192.168.50.154)(PORT=1521))(CONNECT_DATA=(SERVICE_NAME=XE))); uid=system;pwd=root;
EPC Parameter	ControlLogic.EPCParameterName	Define the name used for the EPC within the insert statement. E.g.: :EPC
Location Parameter	ControlLogic.LocationParameterName	Define the location name used for the device within the insert statement. E.g.: :Location
Read Date Parameter	ControlLogic.ReadDateParameterName	Define the name used for the date when the tag is read within the insert statement. E.g.: :ReadDate
SQL Statement	InsertID.Query	Define the insert statement that is valid for your used database. The following example shows a valid insert statement using a MySQL database. The parameters used must be written in the same way as listed in the parameters above. E.g.: INSERT INTO Tags (EPC ,READDATE, LOCATION) VALUES (:EPC ,to_date(:ReadDate, ' MM/DD/YYYY HH:MI:SS AM') , :Location)

Table 7-7: ORACLE Interface Parameter

7.3 Web Service

Web services are typically application programming interfaces (API) or web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services. In common usage the term refers to clients and servers that communicate over the Hypertext Transfer Protocol (HTTP) protocol used on the web. (http://en.wikipedia.org/wiki/Web_service)

The web service is programmed to be failsafe. So an offline buffer is used to fulfill this task.

Offline Buffer All captured data is stored in an offline buffer as long as no web service is registered.

Parameter The following table describes all web service specific parameters.

	Parameter	Description
Date/Time Format	ControlLogic.DateFormat	Define the date and time format string used for the web service.
Offline Buffer Retry	ControlLogic.OfflineBufferRetry	Define the time in milliseconds when the offline buffer will retry to transmit data.

Table 7-8: Web Service Interface Parameter

The **detego[®] EXPRESS** application offers two methods to receive tag data by using a web service.

Modes

The web service can be accessed in two modes:

- Polling Mode
- Event Triggered Mode

7.3.1 Polling Mode

An external application polls the web service of the **detego[®] EXPRESS** application and requests the data of the current tags on the antenna of the RFID reader.

The testing application requires the computer name where **detego[®] EXPRESS** application is running on.

Addon
RegisterURL

Within the **Addon** folder of the **detego[®] EXPRESS** application folder such a test application is provided. This simple .NET application is named **RegisterURL**.

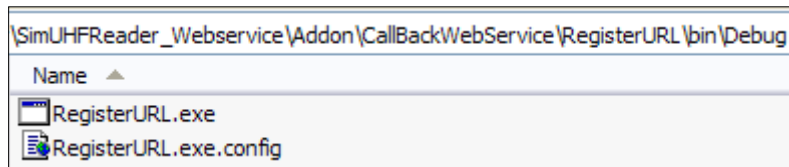


Figure 7-2: RegisterURL Path

RegisterURL.exe.config

If the **detego[®] EXPRESS** application is not running on the same computer, change the computer name within the file *RegisterURL.exe.config*.

Start

Start *RegisterURL.exe*. The first register allows you to test the polling mode.

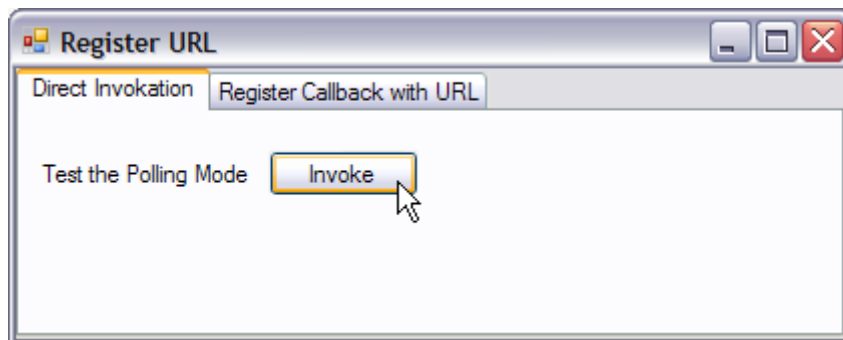


Figure 7-3: RegisterURL Application – Direct Invocation

Click **Invoke** and the currently seen tag is displayed in an information window. E.g.

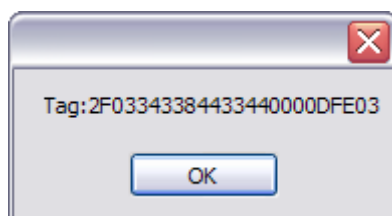


Figure 7-4: Direct Invocation – Tag

7.3.2 Event Triggered Mode

In this case the **detego**[®] **EXPRESS** application must provide a web service that receives the web service events and the external application must register to this web service.

The subfolder `Addon` within the installation folder provides an example how to use the web service.

An installation of Visual Studio 2008 is mandatory to use this example.

Addon
CallBackWebService

Open `CallBackWebService\CallBackWebService.sln` file and run the application from the Visual Studio. The following window will open.

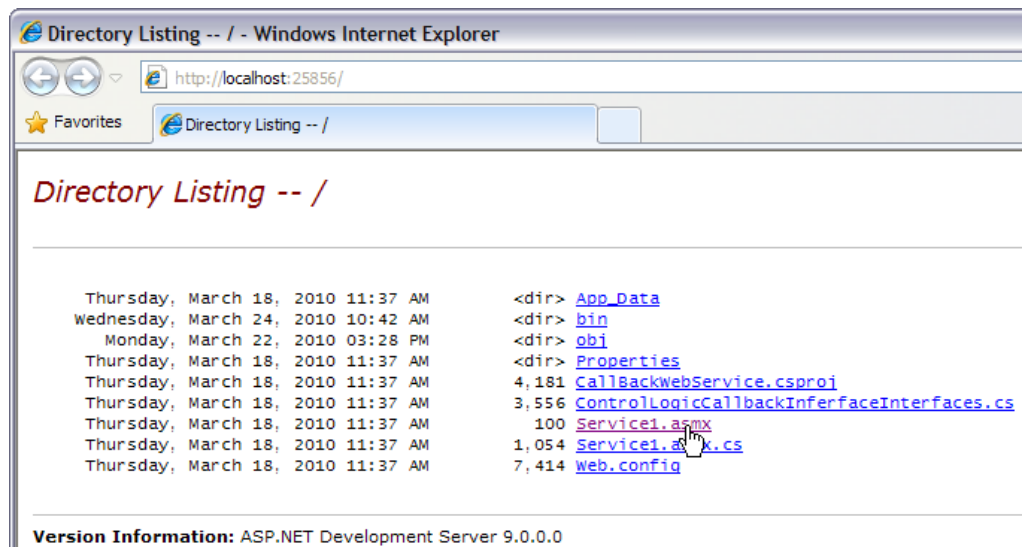


Figure 7-5: CallBackWebService

Click `Service1.asmx`. The following figure shows the result:

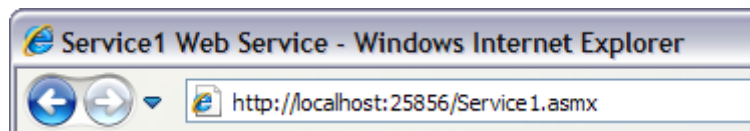


Figure 7-6: Service1.asmx URL

Addon
RegisterURL

Within the **Addon** folder of the **detego**[®] **EXPRESS** application an additional test application is provided. The application is named **RegisterURL**.

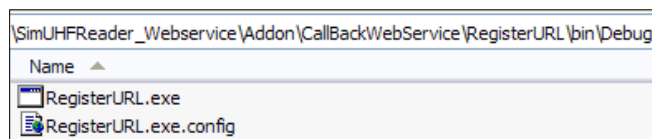


Figure 7-7: RegisterURL Path

RegisterURL.exe.config

If the **detego**[®] **EXPRESS** application is not running on the same computer, change the computer name within the file `RegisterURL.exe.config`.

Start

Start `RegisterURL.exe`. Change to the second register to test the event triggered mode.

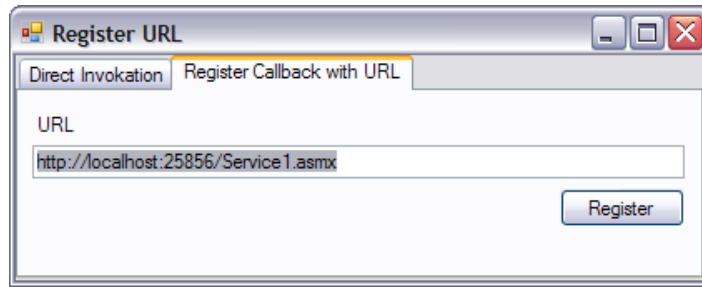


Figure 7-8: RegisterURL Application – Register Callback with URL

Copy the provided URL from the started web service (see Figure 7-6: Service1.asmx URL) as displayed within your browser to the clipboard and paste it into the field URL within the RegisterURL test application.

Start the **detego**[®] **EXPRESS** application and register the web service by clicking the **Register** button.

The data will be written to an output file specified in Service1.asmx.

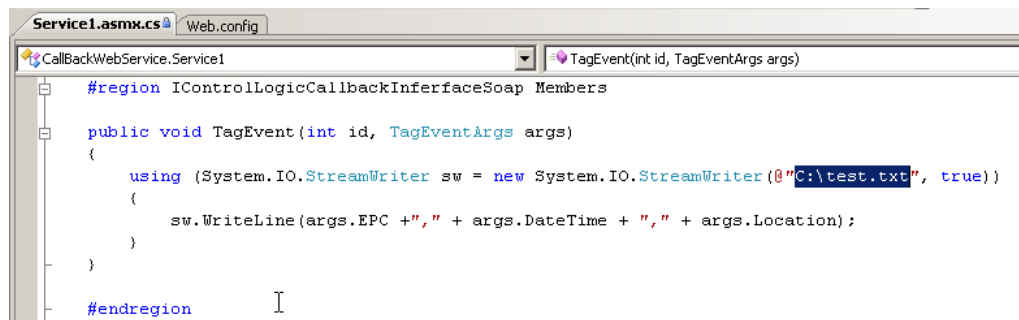


Figure 7-9: Service1.asmx - Output File Definition

Open e.g. C:\test.txt to view the data. The structure is the same as described in Table 7-1: File Structure.

7.4 ALE Web Service

The Ale Web Service is an implementation of the EPCglobal Application Level Events (ALE) Specification 1.0 and provides an interface through which clients may obtain filtered, consolidated EPC data from logical reader sources.

The complete EPCglobal Application Level Events Specification 1.0 can be downloaded from the following URL:

http://www.epcglobalinc.org/standards/ale/ale_1_0-standard-20050915.pdf

An ALE client may access the Web Service via the following URL to define or subscribe to so called Event Cycle specifications (ECSpec):

<http://localhost:9120/WebServices/AleInterface.asmx>

Testclient
Addon\
AleTestClient.exe

detego[®] **EXPRESS** provides a simple ALE client to test the functionality of this package with an ALE Web Service. This test client is located in the folder Addon.

Included are some sample Event Cycle specifications stored in XML (in the ECSpecs folder). To use the client start AleTestClient.exe. You can then start both the ALE test client and the **detego**[®] **EXPRESS** application.

Provided that both the **detego**[®] **EXPRESS** application and the ALE client are running on the same computer, the client will automatically connect to the ALE Web Service. If the **detego**[®] **EXPRESS** application is running on a different machine, you should

change the property **URL** in the **Connection Settings** by replacing “localhost” with the name or IP address of the computer running the **detego® EXPRESS** application.

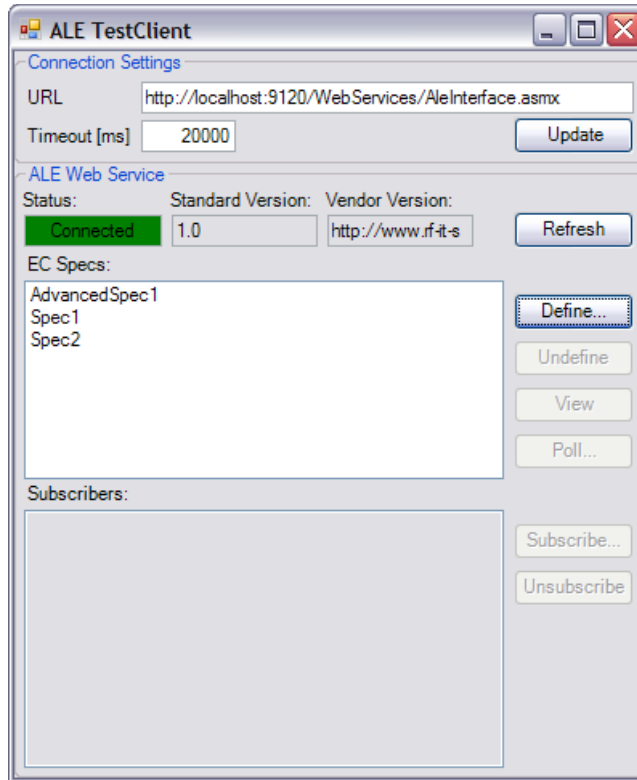


Figure 7-10: ALE Test Client

The test client allows defining and undefining EC specifications, it may be used to synchronously poll reports from these EC Specs and to asynchronously display reports transmitted via HTTP or TCP notifications. In order to subscribe to an EC Spec, one must first be defined using a unique name.

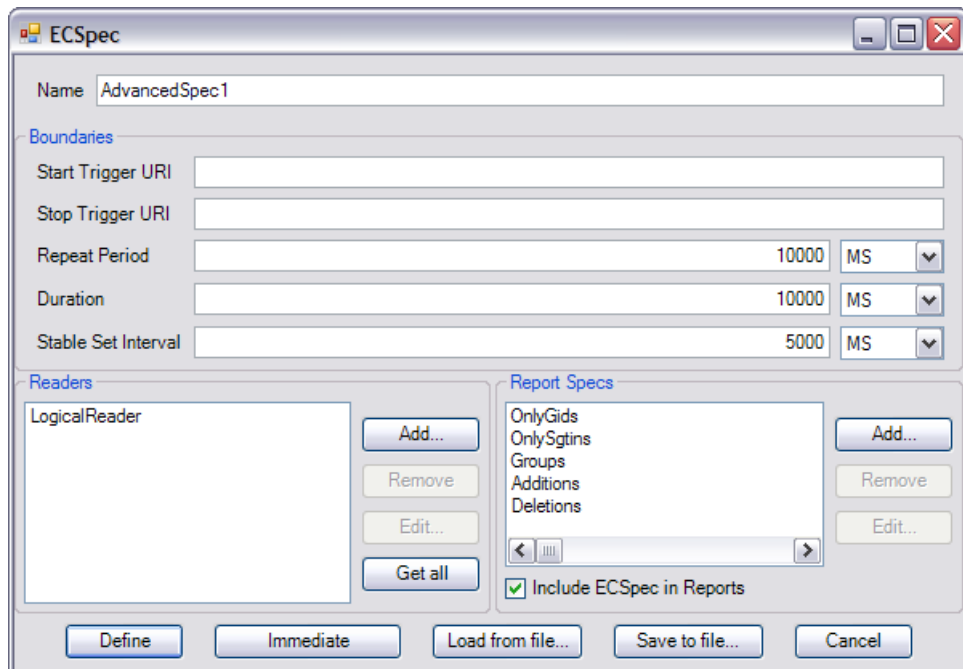


Figure 7-11: Define an EC Spec

By clicking the **Load from file...** button you can load a formerly saved EC spec from its XML representation, e.g. one of the provided sample specifications. Type the name and press **Define** to get the specification. The defined ECSpec can then be subscribed and will transmit notifications to the client.

Figure 7-12: Subscribe an ECSpec



Note: If a failure occurs while subscribing select a different port number e.g. 9999.

The client can receive HTTP and TCP notifications that contain EC reports and displays these reports in a list ordered by the time of their creation.

Figure 7-13: EC Reports

Double-click the entries to view the data stored in the received EC reports.

Group	EPC	Tag	Raw Hex	Raw Decimal
(Default group) [2]				
(Default group)	um.epc.id:sgln:237120.0...	um.epc.tag:sgln:96:0.23...	um.epc.raw:96:x3218E7...	um.epc.raw:96.1550435...
(Default group)	um.epc.id:sgln:227869.2...	um.epc.tag:sgln:195:0.2...	um.epc.raw:200:x31918D...	um.epc.raw:200.358404...

Figure 7-14: Detailed Data of an EC Report

The EC reports contain information about the all EPCs read in the corresponding Event Cycle. Depending on the EC specification, EPCs are assigned to different reports and may be grouped according to the formerly defined filtering rules.

Closing the notifications window will automatically unsubscribe the corresponding EC spec. When all subscribers have been unsubscribed, the EC spec may also be undefined.

Parameter No specific parameters must be set for using an ALE web service.

7.5 HTTP Request

Hypertext Transfer Protocol (HTTP) is a request/response standard typical of client-server computing. In HTTP, web browsers or spiders typically act as clients, while an application running on the computer hosting the web site acts as a server. The client, which submits HTTP requests, is also referred to as the *user agent*. The responding server, which stores or creates *resources* such as HTML files and images, may be called the *origin server*. In between the user agent and origin server may be several intermediaries, such as proxies, gateways, and tunnels.
(<http://en.wikipedia.org/wiki/HTTP>)

Install Install any web server on your desktop PC.

Add-On
TagAdded.php An example PHP script that can be installed on this web server is provided in the subfolder *Addon* within the installation folder. Every time the TagAdded event is fired by **detego® EXPRESS** a HTTP request is sent to the TagAdded.php script. This script writes the received data to the file *tags.txt* within this folder

...\Addon\TagAdded.php

Parameter The following table describes all HTTP request specific parameters.

	Parameter	Description
URL	ControlLogic.URL	Define the URL to the HTTP Request. Parameter with leading @ are replaced with the corresponding values. <code>http://localhost:8080/tagadded.php?tag=@Tag&date=@Date&location=@Location</code>
Date/Time Format	ControlLogic.DateFormat	Define the date and time format string using for the HTTP request.
Offline Buffer Retry	ControlLogic.OfflineBufferRetry	Define the time in milliseconds when the offline buffer will retry to send data.

Table 7-9: HTTP Request Interface Parameter

7.6 Microsoft Message Queuing (MSMQ)

MSMQ is essentially a messaging protocol that allows applications running on separate servers/processes to communicate in a failsafe manner. A queue is a temporary storage location from which messages can be sent when conditions permit. This enables communication across heterogeneous networks and between computers which may not always be connected.

Install

To send messages with tag IDs to a MSMQ, providing you have installed Message Queuing via **Add or Remove Programs|Add/Remove Windows Component**.

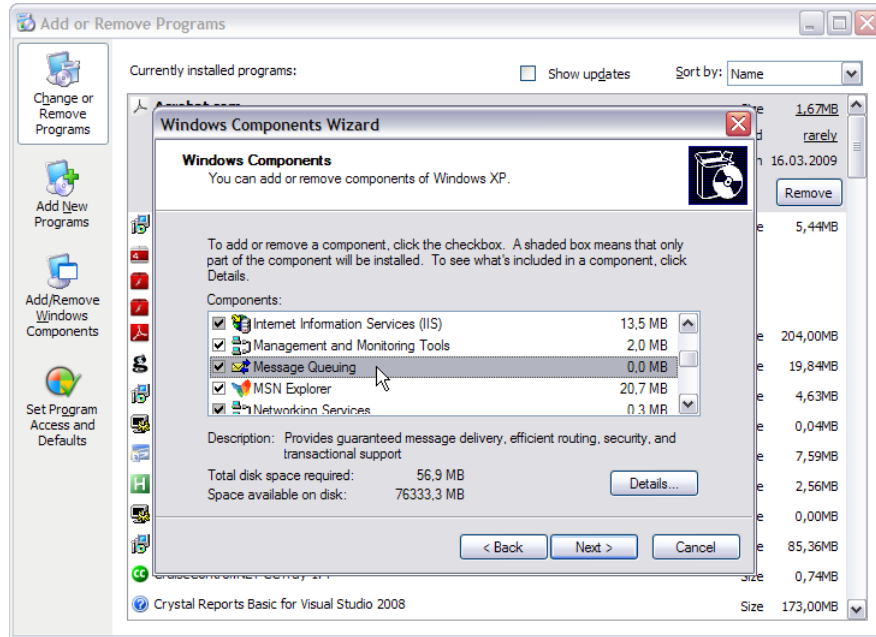


Figure 7-15: Add Window Component Message Queuing

The next step is to create for example a private queue named "testqueue". Select **Control Panel|Administrative Tools|Computer Management** and add a new queue.

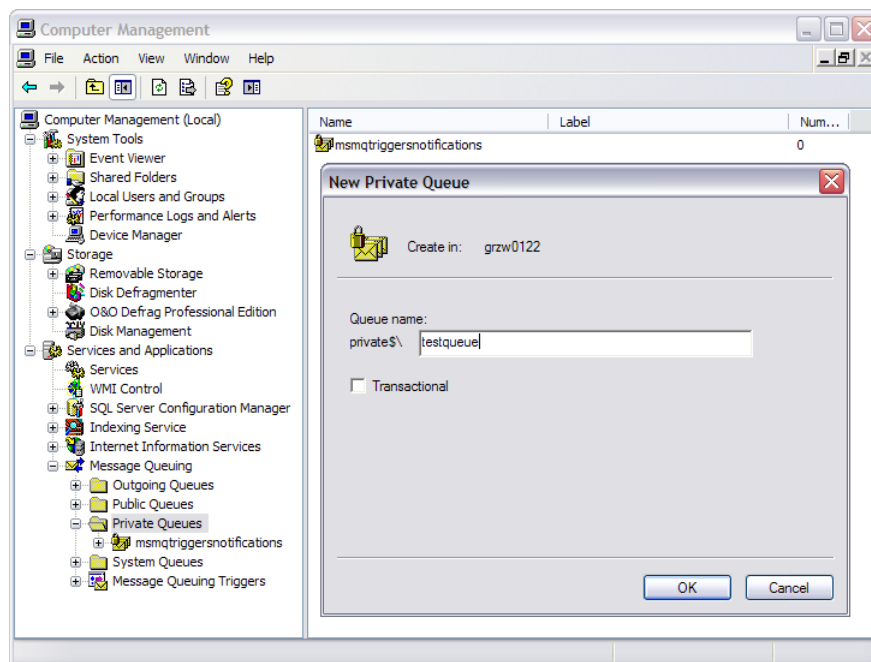


Figure 7-16: Create Private Queue "testqueue"

Parameter

The properties of the created queue give you the value for the **ControlLogic.MSMQ** parameter in **detego® EXPRESS** Configuration tool.



Figure 7-17: "testqueue" Properties

The following table describes all MS Message Queue Connector specific parameters.

	Parameter	Description
MessageQueue	ControlLogic.MSMQ	Name of the message queue in format "computer name\Queue type\Queue name" (e.g. "GRZW0122\private\$\testqueue")

Table 7-10: MSMQ Interface Parameter

Output

When running the **detego® EXPRESS** application the **Number of Messages** of this queue increases when reading a new tag.


Name	Label	Number of Messages
 testqueue	private\$\testqueue	37

Figure 7-18: Number of Messages

Data Structure

The data object used to queue the data into the MSMQ is defined as follows:

```
public ReadTag(string epc, string location, DateTime readDate)
{
    this.Epc = epc;
    this.Location = location;
    this.ReadDate = readDate;
}
```

7.7 DA OPC Connector

The DA OPC Connector uses the DA OPC standard for communication between the **detego® EXPRESS** application and PLC devices. Therefore the OPC connector is a regular OPC client.

Preconditions

Before using **detego® EXPRESS** set the following DA OPC variables in your DA OPC server.

Variable	Description
DAOPCConnector.String.EPC	Contains the EPC value
DAOPCConnector.Date.Timestamp	Contains the actual TimeStamp
DAOPCConnector.String.Location	Contains the Location setting of the file entities.xml

Table 7-11: OPC Connector Variable Definitions

Set the following variables within your DA OPC server according to your requirements:

Variable	Description
DAOPCConnector.Bool.Mode	True: values (epc, timestamp, location) are set automatically on tag added event; False: no values are set
DAOPCConnector.Bool.Trigger	True: first tag on the antenna is used and values are set; False: no data are read

Table 7-12: DA OPC Connector Interface Parameter

Parameter The following table describes all DA OPC Connector specific parameters.

	Parameter	Description
Server node	DAOPCConnector.ServerNode	The network address of the OPC server node.
Server ProgID	DAOPCConnector.ServerProgId	The ProgID of the OPC server.

Table 7-13: DA OPC Connector Interface Parameter

7.8 Process

Using the Process interface it is possible to transfer tag data to any process you like.

Example The supplied example transfers the EPC of the read tag to Firefox web browser and uses it for google search string. So every time a tag is read a new tab opens within the web browser with the EPC as search string.

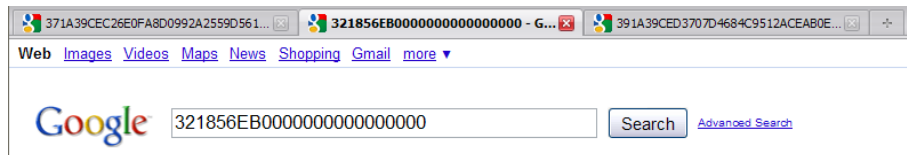


Figure 7-19: Process Interface Example

Addon Within the Addon folder you will find an additional example how to transfer tag data using twitter messages.

Parameter The following table describes all process specific parameters.

	Parameter	Description
CommandPath	ControlLogic.CommandPath	Path to the command tool (e.g. Firefox.exe) Note: When using IExplore.exe for each tag a new window will open!
Arguments	ControlLogic.Arguments	Arguments of the command (e.g. www.google.com/search?q=@Epc)
EPC Parameter	ControlLogic.EPCParameterName	Define the name used for the EPC within the insert statement. E.g.: @Epc
Location Parameter	ControlLogic.LocationParameterName	Define the location name used for the device within the insert statement. E.g.: @Location
Read Date Parameter	ControlLogic.ReadDateParameterName	Define the name used for the date when the tag is read within the insert statement. E.g.: @ReadDate

Table 7-14: Process Interface Parameter

8 Three Additional Ways to Start detego® EXPRESS

Normally **detego® EXPRESS** is started via the graphical interface application as described in chapter 4.2. However, there are three additional ways to start **detego® EXPRESS** via the following programs:

- Log Viewer
- Console Application
- Windows Service

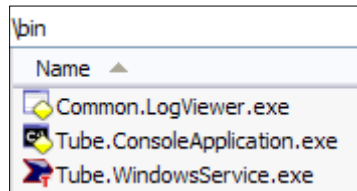


Figure 8-1: Additional detego® EXPRESS Start Applications

8.1 Log Viewer

The Log Viewer application allows comfortable reading of log files by filtering log messages with the use of the **Level** and **Logger** drop-down boxes. Select your preferred log entry to get detailed information shown in the **Details** window.

Start
Common.LogViewer.exe

Start the Log Viewer application by executing the file `Common.LogViewer.exe` within the `bin` folder of the **detego® EXPRESS** application.

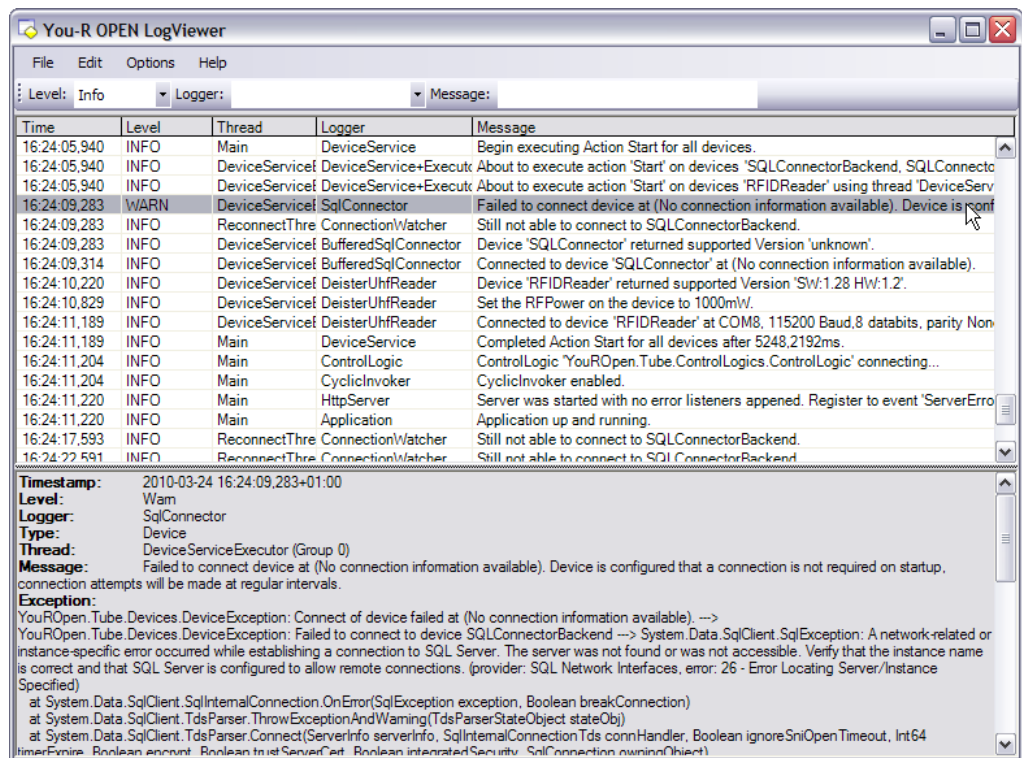


Figure 8-2: You-R® OPEN Log Viewer

Use the **Level** drop-down box to filter log messages that are thrown by the Tube.

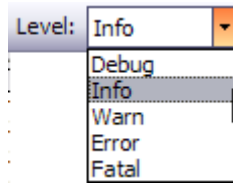


Figure 8-3: Filter Messages Using the Log Level

The **Logger** drop-down box allows filtering messages based on a specific component of **detego® EXPRESS**.

Type part of the message within the **Message** textbox to filter for the message text.

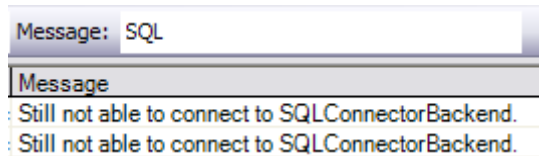


Figure 8-4: Message Filter

The **Details** window at the bottom of the Log Viewer application provides additional information about single entries in the log list.

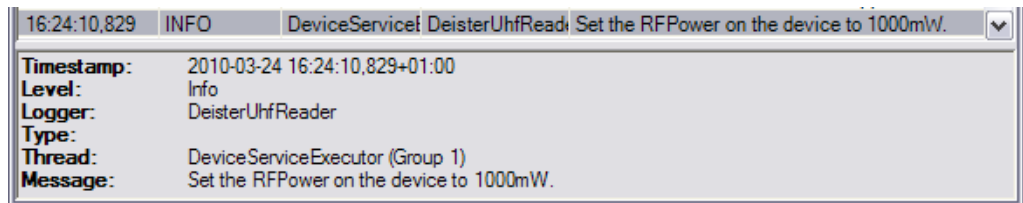


Figure 8-5: Detailed Description of a Log Message

8.2 Console Application

The Console Application outputs the same information as the Log Viewer but displays the information solely inside the console. By doing so, the Console Application achieves a better performance, compared to the Log Viewer application.

Start
Tube.Console
Application.exe

Start the Console application by executing the file `Tube.Console Application.exe` within the `bin` folder of the **detego® EXPRESS** application.

The console application starts your **detego® EXPRESS** application and generates an output like the following:

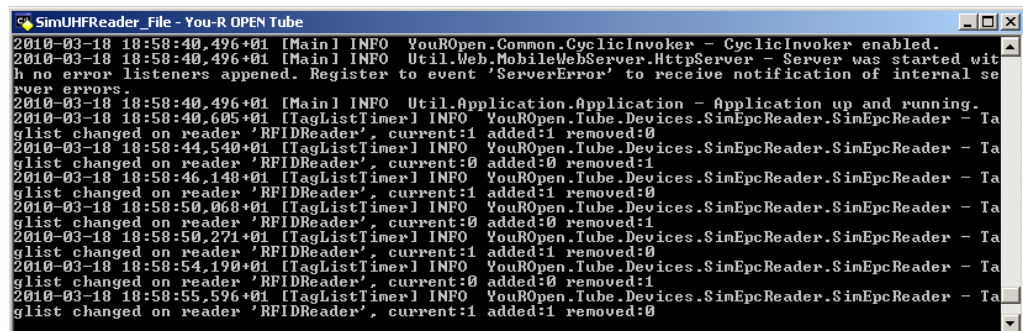


Figure 8-6: Console Application

8.3 Windows Service

It is also possible to run a **detego® EXPRESS** application as a windows service. The advantages of a windows service are:

- the service runs in the background
- the operating system starts the windows service before the GUI is started
- the service can be configured to run with any other user context

Install

Tube.WindowsService.exe

The Windows Service must be installed first. This is done using the parameter `/i`. Start the `Tube.WindowsService` application within the command line using the following command:

```
...\bin>Tube.WindowsService.exe /i
```



Note: Be sure that the path inside the command line points to the `bin` folder of the **detego® EXPRESS** application.

After having registered **detego® EXPRESS** as a Windows service, it can be started and stopped using e.g. the Windows Services application.

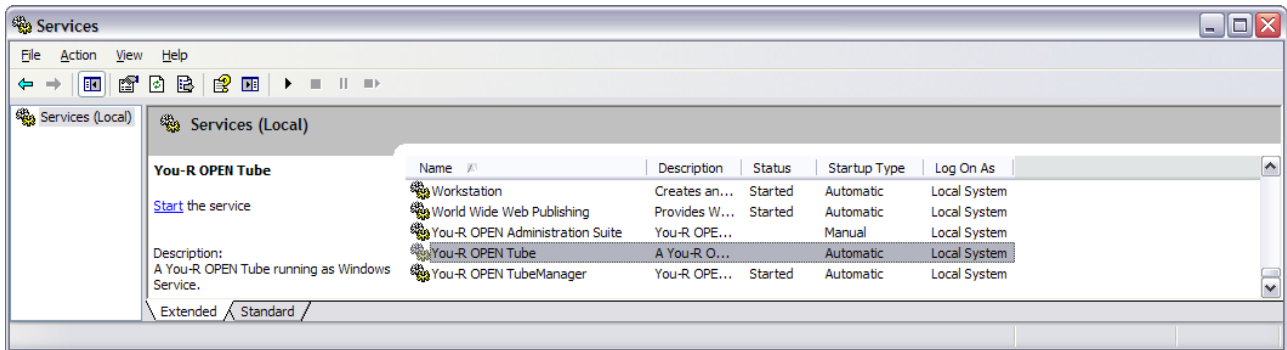


Figure 8-7: Started Windows Service

Uninstall

Uninstall the service via the command line using the `/u` parameter

```
...\bin>Tube.WindowsService.exe /u
```

9 Tools

The following tools will assist you in receiving more tag information and/or widen your insight into **detego® EXPRESS** main functionality.

9.1 You-R® OPEN Verification Client

Sometimes it is wise to check and recheck data of RFID transponders. **detego® EXPRESS** provides the You-R® OPEN Verification Client to fulfill exactly this purpose.

The You-R® OPEN Verification Client is a tool that

- provides a simple visualization of a transponder list which is currently on an antenna field
- allows to activate or deactivate the antenna field
- can store sample data and transponder lists to a file for later use
- can check transponders against specified reference lists



Note: This tool is only available as a separate download link at the RF-iT Solutions Homepage.

With the You-R® OPEN Verification Client you can quickly

- verify and
- change

data on transponders.

The You-R® OPEN Verification Client receives data using an EPC Web Service Interface as described in chapter 10.1.

Download

Download You-R® OPEN Verification Client as a zipped file directly from the following provided link:

<http://www.rf-it-solutions.com/en/produkte/detego-express.html>

Installation

After extracting the files you will find the following file structure on your computer:

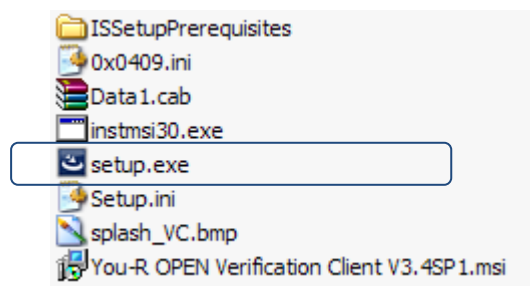


Figure 9-1: You-R® OPEN Verification Client File Structure

setup.exe

Double click the file **setup.exe** to install the You-R® OPEN Verification Client. The installation wizard will guide you through the installation process. After successful installation you will find the application with the path displayed in the figure below within the Start menu.

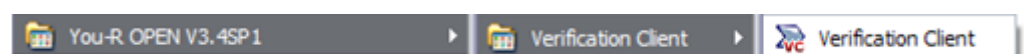


Figure 9-2: Verification Client Installation

Start

When starting the You-R[®] OPEN Verification Client you are prompted for the Tube Interface. The term **Tube** is used as a synonym for **detego[®] EXPRESS** application in this context.



The You-R[®] OPEN Verification Client collects data from the **detego[®] EXPRESS** application. Therefore it is important that the application is running before starting the You-R[®] OPEN Verification Client.

After starting the application the following dialog will open to establish a connection to **detego[®] EXPRESS** application.

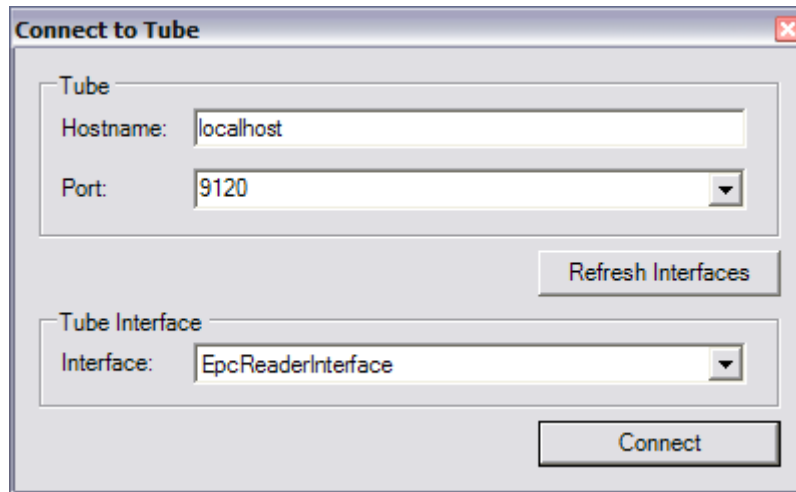


Figure 9-3: Verification Client - Connect to Tube

Click **Connect** using the EpcReaderInterface.



Note: If the **Interface** combo box is grayed out try to **Refresh Interfaces** by clicking the corresponding button. If it remains grayed out check if your **detego[®] EXPRESS** application is running.

The figure below gives you a glimpse of this tool.

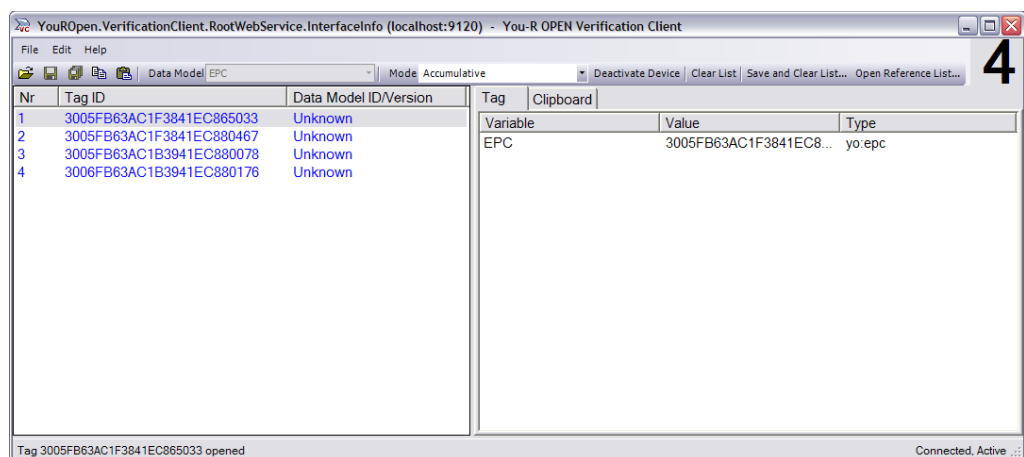


Figure 9-4: Verification Client – Example

Modes

You can switch between two display modes:

- Normal: only transponders actually read are displayed
- Accumulative: all read transponders remain displayed

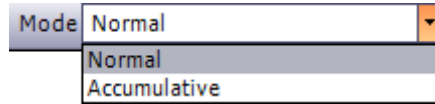


Figure 9-5: Verification Client – Mode Selection

9.1.1 Presentation

Left Side

The left side lists all transponders according the selected mode.

Right Side

The right side of the application gives you the possibility to change values. **detego® EXPRESS** only provides the change of the EPC value.



Note: The EPC value is the same value as displayed in column ID within the **detego® EXPRESS** application.



detego® EXPRESS does not support user memory (data models). To use this feature our advanced development platform product **detego® You-R OPEN** is necessary. Send a request to [RF-iT Solutions](#) to receive more information about [detego® You-R OPEN](#).

Modify EPC Value

To change the EPC value click into the value field and change the value. When leaving the field the value is stored at the transponder.

Tag	Clipboard	
Variable	Value	Type
EPC	3005FB63AC1B3941EC880078	yo:epc

Figure 9-6: Verification Client – Modify EPC Value

Check the modified value by removing the transponder from the antenna field for several seconds and by reentering the transponder to the field. Now the new value should be displayed.

9.1.2 Functionality

The You-R® OPEN Verification Client provides several functionalities.



Figure 9-7: Provided Functionalities

9.1.2.1 Activate / Deactivate Device

Within the You-R® OPEN Verification Client you can activate or deactivate the RFID device. Click the toggling button **Activate Device** / **Deactivate Device** to fulfill this task.

9.1.2.2 Tag Lists



Note: When you plan to work with lists, change to the **Accumulative** mode. All tags seen on the antenna field are listed now.

Nr	Tag ID	Data Model ID/Version
1	3005FB63AC1F3841EC865033	Unknown
2	3005FB63AC1F3841EC880467	Unknown
3	3005FB63AC1B3941EC880078	Unknown
4	3006FB63AC1B3941EC880176	Unknown
5	34680000000000000000000010	Unknown
6	3005FB63AC1F3841EC880468	Unknown

Figure 9-8: List displayed in You-R® OPEN Verification Client

Clear List

To clear the populated list, click the **Clear List** button.

Save and Clear List...

Click the button **Save and Clear List...** to save the list seen at the left side of the application to a CSV file and clear the displayed list.

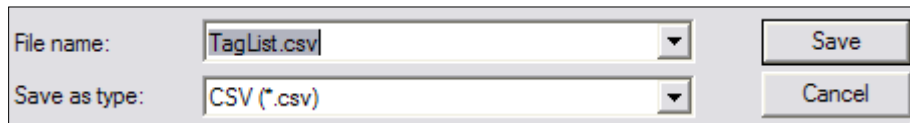


Figure 9-9: Save and Clear List – Save As Dialog

The output file will be structured like this:

```
EPC;TagId
3005FB63AC1F3841EC880468;3005FB63AC1F3841EC880468
3468000000000000000000000010;3468000000000000000000000010
3005FB63AC1F3841EC865033;3005FB63AC1F3841EC865033
3005FB63AC1B3941EC880078;3005FB63AC1B3941EC880078
```

Figure 9-10: Content of the File TagList.csv

The **TagId** as displayed at the left side of the You-R® OPEN Verification Client and the variable **EPC** are stored to the CSV file.

Open Reference List...

A useful functionality is to check EPCs against a reference list. This list must be structured as displayed in Figure 9-10: Content of the File TagList.csv.

First the You-R® OPEN Verification Client displays all Tag IDs read from the reference list in red color. When a tag from the reference list is read the correspondent entry changes its color to green. Blue colored lines identify read tags that are not found inside the reference list

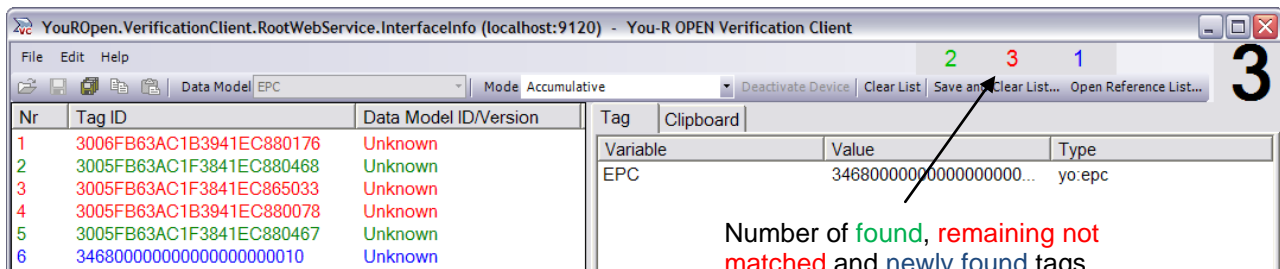


Figure 9-11: You-R® OPEN Verification Client – Scan against Reference List

9.2 EPC GEN2 Application

Use the EPC GEN2 Application to see additional information stored on the transponder.

The following values can be checked:

- EPC (can also be modified within this application)
- EPC type
- EPC values corresponding to the type
- Tag type
- Read time
- Tag ID
- RSSI (Received Signal Strength Indication)



Note: **detego**[®] **EXPRESS** does not support the locking and killing password functionality.

Start
Tube.EPCGen2
Application.Gui.exe

Start the EPC GEN2 application by executing the file `Tube.EPCGen2 Application.Gui.exe` within the `bin` folder of the **detego**[®] **EXPRESS** application.

The figure below gives you a glimpse of the EPC GEN2 application.

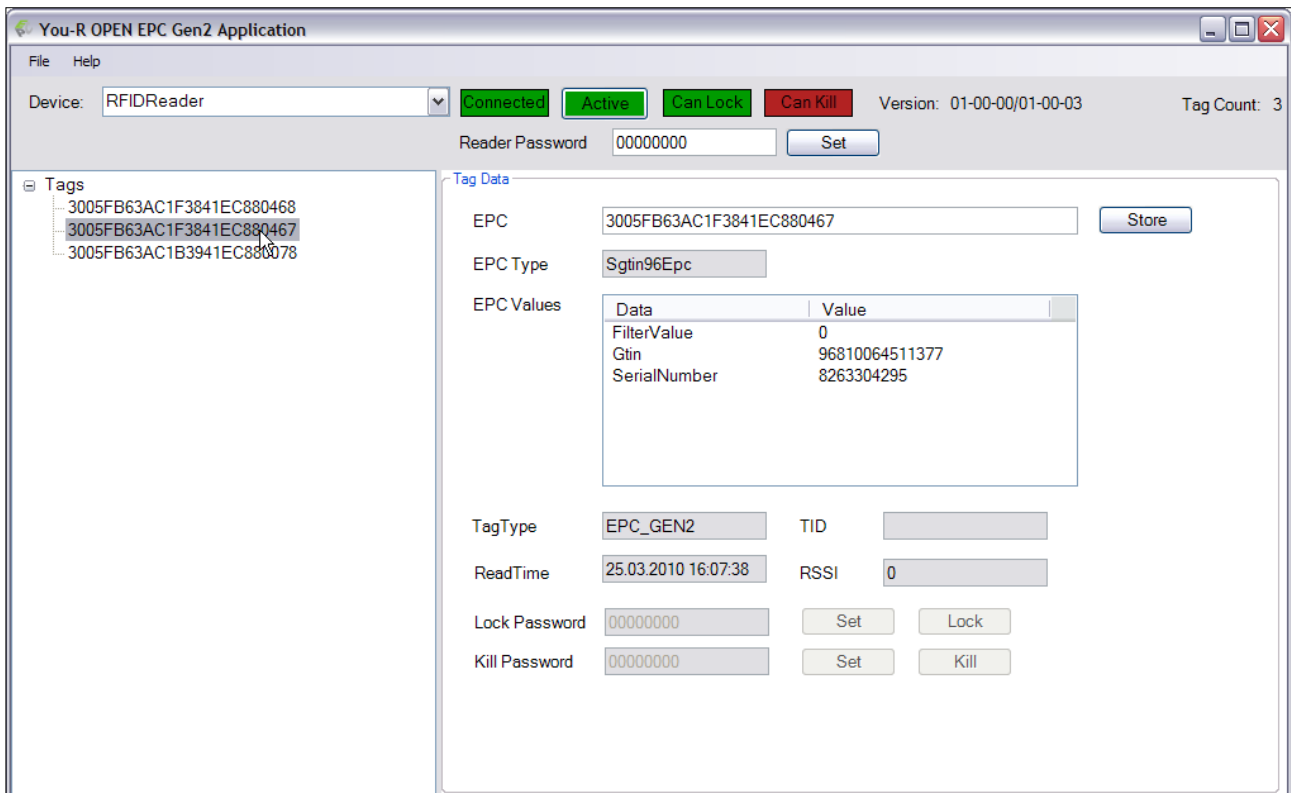


Figure 9-12: EPC GEN2 Application

Depending on the device some information is displayed:



Figure 9-13: Device Information

The following states can be checked:

- connection state
- device is able to lock the password
- device is able to kill the password
- firmware-version number of the device

Actions:

- device can be deactivated/activated
- set a password for the reader
- store a new EPC on the tag

10 Enhanced Web Service Interface

10.1 EPC Reader Web Service

When a **detego**[®] **EXPRESS** application is running (GUI application, Log Viewer, Console Application or Web Service), you can browse to the WSDL file typing <http://localhost:9120> within your browser.

Addon
EPC Webservice.zip

An example application is available inside the folder `Addon` as the zipped file: `...\Addon\EPC Webservice.zip`

After extracting the zipped file you will find a file `readme.txt` that gives you further information on how to use this example.

This example application allows you to

- register three events (TagAdded, Removed, Stored)
- get the tag list
- activate or deactivate the reader
- retrieve the activation and connection status
- store, lock or kill tags, if provided by the reader

The following figure demonstrates the example test application. Navigate to `...\Addon\EPC Webservice\CallBackWebService\EPCReaderServiceApp\bin\Debug\RegisterURL.exe`, to start this application.

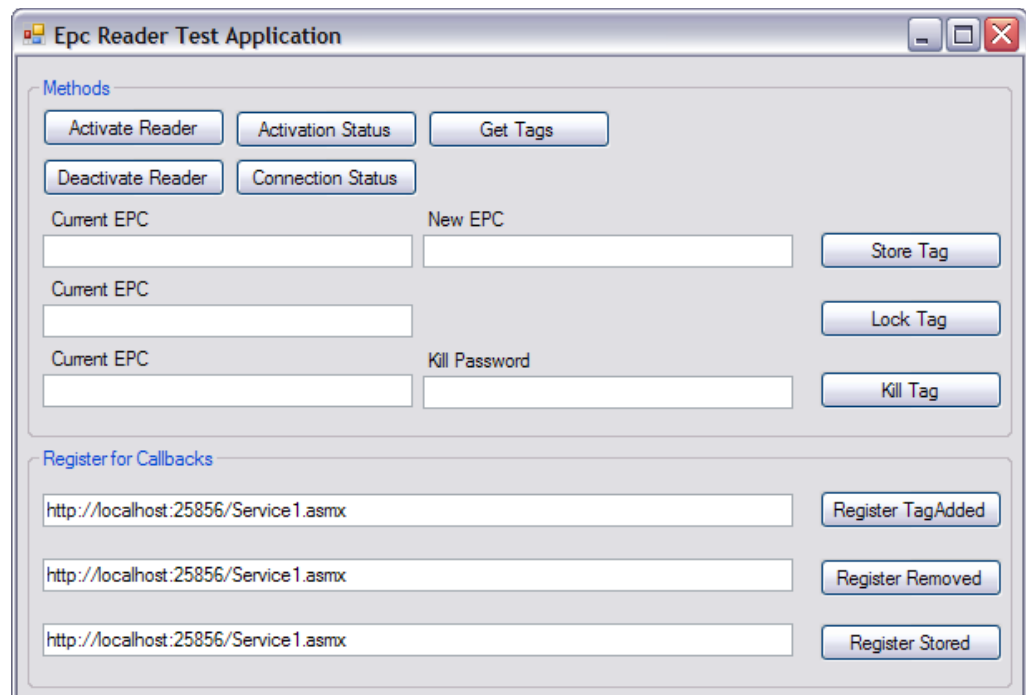


Figure 10-1: EPC Reader Test Application

10.2 Facade .Net

The Facade Design Pattern provides a simple interface and controls access to a series of complicated interfaces and or sub systems.

Addon
Facade .Net
Example.zip

An example application is available inside the folder `Addon` as the zipped file: `...\Addon\Facade .Net Example.zip`

After extracting the zipped file you will find a file `readme.txt` that gives you further information on how to use this example.

This is a simple example to show how to develop your own application based on the **detego[®] EXPRESS** application. This will be possible using all .Net programming languages.

A few steps are necessary:

- Open the Visual Studio 2008 solution file: `Facade.sln`
- Add the references of `Util.dll`, `Common.dll` and `Tube.dll` from your **detego[®] EXPRESS** applications `bin` folder.
- Build the solution
- Copy the builded `...\Facade\bin\Debug\Facade.exe` to the `bin` folder of your **detego[®] EXPRESS** application.
- Start `Facade.exe`

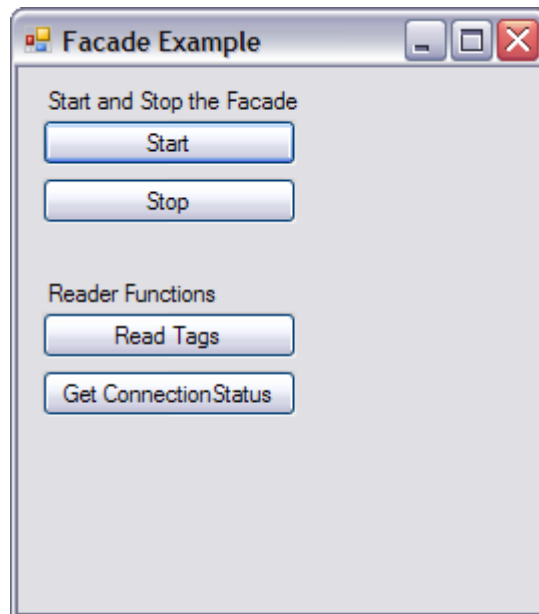


Figure 10-2: Facade .Net Example Application

11 Glossary

Backend System

Backend systems loosely refer to servers, super servers, clustered systems, midrange systems, and mainframes that provide data services to users. Users interact with applications in front-end systems to make requests on back-end systems. The back-end systems then process the requests, searching and sorting data, serving up files, and providing other services. Back-end systems are physically close to data storage systems, so this arrangement uses the network efficiently.

Barcode

A barcode is an optical machine-readable representation of data, which shows certain data on certain products. Originally, barcodes represented data in the widths (lines) and the spacings of parallel lines, and may be referred to as linear or 1D (1 dimensional) barcodes or 2D barcodes.

Control Logic

A Control Logic represents a single RFID process. Within one Tube multiple Control Logics are able to communicate with each other. For example, a Control Logic is able to transfer barcodes into EPCs. Other devices, connectors or references can be accessed and modified as well.

Data Model

A data model is a set of variable types describing user memory.

detego®

detego® is synonymous with leading RFID solutions along the entire supply chain and sets standards in terms of flexibility, scalability, technology and functionality.

detego® You-R OPEN

detego® You-R OPEN is an advanced software product that allows you to connect, configure and control a wide range of Auto ID devices, and interface to a database or an Enterprise Application (e.g. SAP). Most of the leading devices are supported including barcode scanners, mobile computers, printers, RFID (Radio Frequency Identification) readers and I/O devices. It includes a library of pre-configured templates (referred to as Tubes), an Administration Suite (for managing devices), a Tube Builder (for developing new solutions) and a Verification Client (for testing).

detego® You-R OPEN provides a cost-effective software backbone that allows systems integrators to deliver a wide range of barcode and RFID solutions – quickly and easily. The Tube libraries and market-specific versions enable systems integrators to develop ‘repeatable’ solutions with minimal configuration – saving time and money.

Device

A device is a general term for hardware and its corresponding software driver within the Tube which communicates with the hardware. The functionality of the hardware can be accessed within a Control Logic. (RFID reader, digital I/O, barcode scanner, etc.)

EPC

The EPC (electronic product code) is an electronically recorded code on an RFID tag. The EPC is a 64- or 96-bit code, which is based on current numbering schemes. It is divided into numbers that differentiate the product and manufacturer of a given item.

ERP

Enterprise resource planning (ERP) is an integrated computer-based system used to manage internal and external resources including tangible assets, financial resources, materials, and human resources.

Event

Event (computing), a software message indicating that something has happened

HF

High frequency (HF) radio frequencies of 13,56 MHz.

ID

Number used to identify a tag (e.g. EPC)

Interface

An interface defines a set of functions provided by a You-R® OPEN component. Interfaces can have different transport types, for example SOAP or .NET Remoting. Interfaces are used to provide functionality of You-R® OPEN components with third party software. In contrast to a connector, which depends on the enterprise application it is connected to and which can only be connected to one enterprise application at the same time, an interface provides its functionality to 0 to n third party components, whoever wants to connect and use the interface.

LF

Low frequency or low freq or LF refers to radio frequencies (RF) of 125 kHz. In Europe, and parts of Northern Africa and of Asia, part of the LF spectrum is used for AM broadcasting as the long wave band. In the western hemisphere, its main use is for aircraft beacon, navigation (LORAN), information, and weather systems.

MES

Manufacturing Execution Systems (MES) are information technology systems that manage production in factories.

OPC

The OPC (OPEN Connectivity) standards specify the communication of industrial process data, alarms, events, historical data and batch process data between sensors, instruments, controllers, software systems and notification devices. detego® EXPRESS provides an OPC connector.

PCL

A PCL (programmable logic controller) or programmable controller is a digital computer used for automation of electromechanical processes, such as control of machinery on factory assembly lines, amusement rides, or lighting fixtures. PLCs are used in many industries and machines. Unlike general-purpose computers, the PLC is designed for multiple inputs and output arrangements, extended temperature ranges, immunity to electrical noise, and resistance to vibration and impact. Programs to control machine operation are typically stored in battery-backed or non-volatile memory. A PLC is an example of a real time system since output results must be produced in response to input conditions within a bounded time, otherwise unintended operation will result. detego® EXPRESS can link data to PLCs via an OPCServer.

(http://en.wikipedia.org/wiki/Programmable_logic_controller)

RFID

Radio-frequency identification (RFID) is the use of an object (typically referred to as an RFID tag) applied to or incorporated into a product, animal, or person for the purpose of identification and tracking using radio waves. Some tags can be read from several meters away and beyond the line of sight of the reader. Radio-frequency identification comprises *interrogators* (also known as *readers*), and *tags* (also known as *labels*).

RSSI

In telecommunications, received signal strength indicator (RSSI) is a measurement of the power present in a received radio signal.

SCM

Supply chain management (SCM) is the management of a network of interconnected businesses involved in the ultimate provision of product and service packages required by end customers (Harland, 1996). Supply Chain Management spans all movement and storage of raw materials, work-in-process inventory, and finished goods from point of origin to point of consumption (supply chain).

Transponder

Short for Transmitter-responder - an automatic device that receives, amplifies, and retransmits a signal on a different frequency.

Tube

A Tube represents software which manages devices, connectors and interfaces to perform the required process steps defined within Control Logics.

Tube Builder

The You-R® OPEN Tube Builder allows Tube Templates to be developed, using a graphical user interface.

UID

The UID (unique identifier) is used to identify a tag.

UHF

Ultra high frequency (UHF) designates a range of electromagnetic waves with frequencies of 868 MHz in Europe and 910 MHz in USA.

Verification Client

The You-R® OPEN Verification Client is a tool that supports a simple visualization of a transponder list which is currently on an antenna field, allows to activate or deactivate the antenna field, can store sample data and transponder lists to a file for later use, can check transponders against reference lists.

Web Service

Web services are typically application programming interfaces (API) or web APIs that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services.

WMS

A warehouse management system, or WMS, is a key part of the supply chain and primarily aims to control the movement and storage of materials within a warehouse and process the associated transactions, including shipping, receiving, putaway and picking. The systems also direct and optimize stock putaway based on real-time information about the status of bin utilization.

12 Figures

Figure 1-1: detego [®] EXPRESS Architecture	5
Figure 2-1: detego [®] EXPRESS Connections	7
Figure 3-1: detego [®] EXPRESS File Structure	9
Figure 3-2: Windows7 - Run as administrator	9
Figure 3-3: Running detego [®] EXPRESS Application	10
Figure 3-4: Connected Information	11
Figure 3-5: Device - Not Connected	11
Figure 3-6: Tag List.....	11
Figure 3-7: File Menu.....	12
Figure 3-8: Close Window (Application)	12
Figure 4-1: Help Menu - License	13
Figure 4-2: Open license.yol.....	13
Figure 5-1: About Dialog.....	14
Figure 6-1: detego [®] EXPRESS Configuration Tool – Example.....	15
Figure 6-2: Parameter Group Name	15
Figure 6-3: Parameter Representation	15
Figure 6-4: Changing Boolean Variables (unchecked box means false)	16
Figure 6-5: Save Configuration Values.....	16
Figure 7-1: ODBC Data Source Definition.....	21
Figure 7-2: RegisterURL Path	24
Figure 7-3: RegisterURL Application – Direct Invocation	24
Figure 7-4: Direct Invocation – Tag	24
Figure 7-5: CallBackWebService.....	25
Figure 7-6: Service1.asmx URL	25
Figure 7-7: RegisterURL Path	25
Figure 7-8: RegisterURL Application – Register Callback with URL.....	26
Figure 7-9: Service1.asmx - Output File Definition.....	26
Figure 7-10: ALE Test Client	27
Figure 7-11: Define an ECSpec.....	27
Figure 7-12: Subscribe an ECSpec	28
Figure 7-13: EC Reports.....	28
Figure 7-14: Detailed Data of an EC Report	28
Figure 7-15: Add Window Component Message Queuing	30
Figure 7-16: Create Private Queue "testqueue"	30
Figure 7-17: "testqueue" Properties	31
Figure 7-18: Number of Messages	31
Figure 7-19: Process Interface Example	32
Figure 8-1: Additional detego [®] EXPRESS Start Applications	33
Figure 8-2: You-R [®] OPEN Log Viewer	33

Figure 8-3: Filter Messages Using the Log Level 34

Figure 8-4: Message Filter 34

Figure 8-5: Detailed Description of a Log Message 34

Figure 8-6: Console Application 34

Figure 8-7: Started Windows Service 35

Figure 9-1: You-R® OPEN Verification Client File Structure 36

Figure 9-2: Verification Client Installation 36

Figure 9-3: Verification Client - Connect to Tube 37

Figure 9-4: Verification Client – Example 37

Figure 9-5: Verification Client – Mode Selection 38

Figure 9-6: Verification Client – Modify EPC Value 38

Figure 9-7: Provided Functionalities 38

Figure 9-8: List displayed in You-R® OPEN Verification Client 39

Figure 9-9: Save and Clear List – Save As Dialog 39

Figure 9-10: Content of the File TagList.csv 39

Figure 9-11: You-R® OPEN Verification Client – Scan against Reference List 39

Figure 9-12: EPC GEN2 Application 40

Figure 9-13: Device Information 40

Figure 10-1: EPC Reader Test Application 42

Figure 10-2: Facade .Net Example Application 43

13 Tables

Table 3-1: Tag Information 11

Table 6-1: Parameter Types 15

Table 6-2: General Parameter 16

Table 6-3: Serial Connection Parameter 17

Table 6-4: TCP Connection Parameter 17

Table 6-5: Distinct Filter Parameter 18

Table 6-6: EPC Pattern Filter Parameter 18

Table 7-1: File Structure 19

Table 7-2: File Interface Parameter 19

Table 7-3: OLE DB Interface Parameter 20

Table 7-4: ODBC Interface Parameter 21

Table 7-5: MySQL Interface Parameter 22

Table 7-6: MS SQL Server Interface Parameter 22

Table 7-7: ORACLE Interface Parameter 23

Table 7-8: Web Service Interface Parameter 23

Table 7-9: HTTP Request Interface Parameter 29

Table 7-10: MSMQ Interface Parameter 31

Table 7-11: OPC Connector Variable Definitions	31
Table 7-12: DA OPC Connector Interface Parameter	32
Table 7-13: DA OPC Connector Interface Parameter	32
Table 7-14: Process Interface Parameter	32