



AGETOR®

AXT Tracking
User Guide

Table of contents

1	Overview.....	3
2	Installation.....	5
2.1	Properties	5
2.1.1	Database properties	6
2.1.2	Event handling properties	6
2.1.3	Database cleanup properties.....	7
2.2	Database setup.....	8
2.2.1	Using the MySQL database in a new installation.....	8
2.2.2	Important – converting data from AXT Tracking 1.0.x.....	8
2.2.3	Important – updating the database from AXT Tracking 2.0.0.....	9
3	Configuration	10
3.1	Documents - tracking.xml.....	10
3.2	Events - tracking_event.xml	15
3.3	Configuration – Filter set up	17
3.4	Use of non XML-documents in AXT Tracking.....	18
3.4.1	Exceptions.....	19
4	AXT Tracking administration site	20
4.1	Simple search	20
4.2	Sorting.....	21
4.3	Writing log to a reference id	21
4.4	Reference detail.....	21
4.5	Closing / Deleting references.....	22
4.6	Advanced Search.....	22
4.7	Choose the number of shown reference.....	23
4.8	Current Status.....	23
4.9	Refreshing	23
5	Configuration problems	24

1 Overview

AXT Tracking monitors the flow of documents in AXT, extracts information and stores this information in a database. Different types of business documents can be grouped together, thus enabling monitoring of a complete business process.

For example, a customer order is received by AXT containing information to be tracked, for instance, the order content, the customer etc. Upon receiving the document Tracking creates a unique reference id and assigns it to the order document. Afterwards the order can be viewed at the Tracking Administration site by locating the reference id.

AXT Tracking adds a *reference id* containing one or more documents with the same reference id. The reference id could be an order number or some other piece of information identifying the complete business process. Each reference id groups together one or more business documents. For instance, documents related to the above-mentioned order could be an order confirmation or an invoice. In AXT Tracking all three documents and extracted information can be viewed as one business process identified by the reference id.

Another important part of AXT Tracking is the *event handling*. The event handling controls the flow of documents within a business process. It describes which documents that needs to be present at any given time. For instance if a certain document has not arrived within the specified time frame, warnings and errors can be issued telling an administrator that a certain document has not arrived allowing him or her to take action.

AXT Tracking accepts incoming documents in any text form, but the main focus is on XML-documents. The abstract view of the AXT Tracking system is shown below in fig. 1.

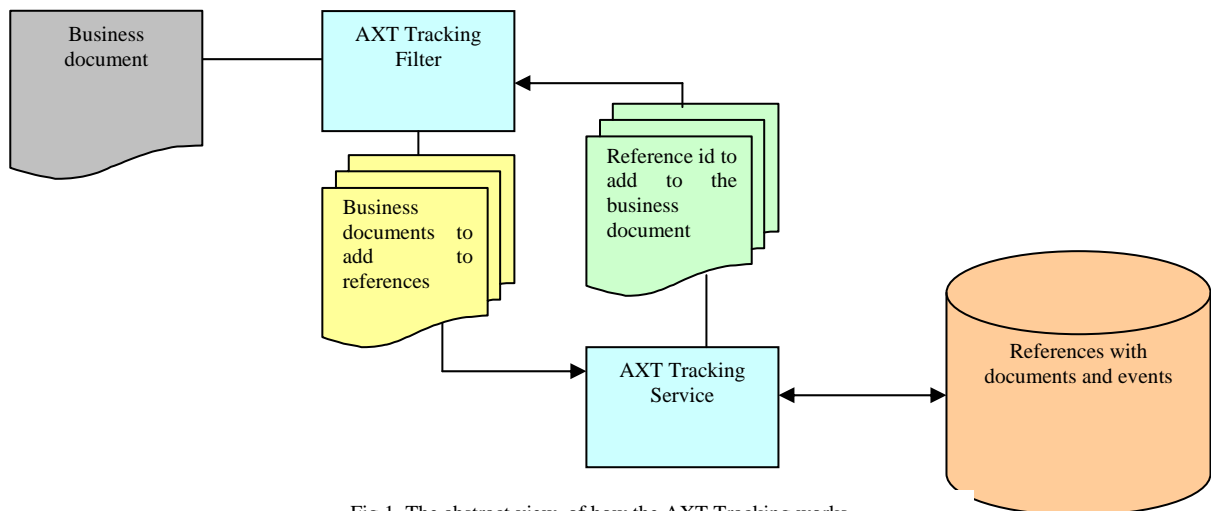


Fig 1. The abstract view, of how the AXT Tracking works.

AXT Tracking has two basic configuration files:

- **tracking.xml**: Sets up incoming documents, which information to extract from these and how this information are saved as name value pairs. Thus, this configuration file contains all possible document types and what to extract from them. This is explained in detail in section 3.1.
- **tracking_event.xml**: Sets up the document flow by stating which documents are needed by other documents. It also specifies the maximum amount of time before a certain document must be present - otherwise an event is triggered. This is explained in detail in section 3.2.



It is possible to have documents of the same type configured. For example, an order may not look the same from two different customers. In the “AXTConfiguration.xml”, the AXT Tracking Filter is called with a scope id, to hit the correct document entry in the “tracking.xml”. This is explained in section 3.3.

A major part of AXT Tracking is the visualization of the reference id’s and their documents. This is done at the AXT Tracking Administration web site. Section 4 of this document examines this topic including the available search mechanisms.

Finally this document contains a help section, which explains some of the errors and problems, which could occur. If you have any problems using AXT Tracking check section 5 for a possible solution.

2 Installation

This section describes step-by-step how to install AXT Tracking.

-  The newest version of AXT Tracking is always available at the Bording Data download center at <http://www.agetor.com>.
-  If upgrading from version 1.0.x the installation must first be upgraded to 2.0.0 before continuing with this installation.

Before you begin the installation of AXT Tracking, make sure your installation meets the following prerequisites:

- Java Development Kit (JDK) version 1.3 or newer.
- AGETOR Development Kit (ADK) version 2.0.2 or newer.
- AXT Basic version 2.0.3 or newer.
- A JDBC enabled database. MySQL is the recommended database

Download the newest version of the AXT Tracking and place the package into your “AGETOR_HOME/install/packages” directory. If you are upgrading an existing installation the current configuration is retained.

- Open a command window with “AGETOR_HOME/bin/prompt.bat” and run “installer.bat”.
- After the AGETOR® installtool has started open an Internet browser and point to “http://localhost:8020”.
- If prompted for login and password, please type in your login and password.
- Under “Product(s) ready to install” click on “AXT Tracking 2.x.x” and answer the few questions. Please see section 2.2 for further explanation of the database configuration.
- If this is a new installation, setup your database for use with AXT Tracking. Refer to section 2.2.1 for more information.
- If this is an update of an existing installation you need to update the information stored in the database. If you are upgrading from AXT Tracking 1.0.X you must convert old data into the database, please see section 2.2.2. When updating from AXT Tracking 2.0.0 you will have to update the database, please see section 2.2.3.

Please note, that new properties might have been added. Check out the release.txt.

2.1 Properties

This section describes the properties that are set in the file “tracking.properties”.

The main setting is the path to the document configuration file tracking.xml.

```
# -*-*-*-*- Axt Tracking configuration files -*-*-*-*-  
# The configuration file of the Axt Tracking system  
axt.tracking.configuration.file=${AGETOR_HOME}/conf/axt/tracking/tracking.xml
```

2.1.1 Database properties

AXT Tracking requires a database. The settings stating which database to use, where to find it and how to connect to it are all found in the database configuration part of “tracking.properties”. There are a total of 4 properties. The default settings show how to connect to a MySQL database found on the same computer (localhost). MySQL is the recommended database.

```
# -*-*-*- AXT Tracking DataBase setup -*-*-*-  
  
# The url used to the database  
axt.tracking.db.jdbc.url=jdbc:mysql://localhost/tracking  
  
# The jdbc driver which is used to connect to the database  
axt.tracking.db.jdbc.driver=com.mysql.jdbc.Driver  
  
# The username which to connect to the database  
axt.tracking.db.user.name=  
  
# And the password of this user  
axt.tracking.db.user.password=
```

2.1.2 Event handling properties

There are 7 event-handling properties.

The first property is by far the most important. It denotes the path the file “tracking_event.xml”. This file configures the document-event setup.

The next property tells the system how often it should check for new events. This number is in milliseconds, i.e. 1 second = 1000, 1 minute = 60000 and 1 hour = 3600000. If your system has a lot of open references, the rate should be at least 60000 (1 minute).

The five last settings are the configuration used by the event system to send email and SMS. You must specify which SMTP server to use; the user name and password to use and finally specify which email address the messages are sent from.

If you want to use SMS to send alerts and errors as texts to mobile phones, you must specify the email address to append to the phone number you are sending the SMS to. Please note, that your carrier may not have this option enabled.

```
# -*-*-*- AXT Tracking event handling -*-*-*-  
  
# The configuration file for the event handling.  
axt.tracking.event.configuration.file=${AGETOR_HOME}/conf/axt/tracking/tracking_event.xml  
  
# How often is the reference list checked for events, in milliseconds.  
axt.tracking.event.rate=20000  
  
# The SMTP server mails are send through  
axt.tracking.event.mail.smtphost=mail.somecompany.com  
  
# The SMTP user to send the mail with  
axt.tracking.event.mail.smtuser=user  
  
# The SMTP users password  
axt.tracking.event.mail.smtppassword=password  
  
# What is put in the from field  
axt.tracking.event.mail.smtpfrom=someperson@somecompany.com  
  
# If you want to send a sms, please tell us which email to set after the number  
# Danish Tele Companies :  
# - Sonofon = @note.sonofon.dk  
# - TDK    = @sms.tdk.dk  
axt.tracking.event.sms.compemail=@smsemail.somecompay.com
```

2.1.3 Database cleanup properties

When a reference has been deleted it is kept in the database in inactive-mode. This is done to make sure you don't delete any data by mistake. Only when you are completely sure, the closed references are deleted physically from the database.

It is possible to automatic make the system delete any inactive reference after a specified number of days. You can also specify to save the deleted reference in XML files and keep them as a backup.

There are 3 settings:

```
# Interactive references should be deleted after x days?  
axt.tracking.delete.after=30  
  
# When it is deleted from the database, do you wish to save the reference as a xml-file (true/false)  
axt.tracking.delete.savefile=false  
  
# And where to save the xml file  
axt.tracking.delete.savefile.dir=${AGETOR_HOME}/data/axt/tracking/xml/references
```

2.2 Database setup

To use AXT Tracking you need a JDBC enabled database. It must be configured for use with AXT Tracking. MySQL is the recommended database and the following section is a guide for installing AXT Tracking with MySQL.


2.2.1 Using the MySQL database in a new installation

Download MySQL from <http://www.mysql.com> and install it on a server.

Download the JDBC driver (MySQL Connector/J) from the same site. The package contains a jar file named `mysql-connector-java-xxxxx.jar`. This file must be placed in `$INSTALL_LIB_DIR` (standard environment variable defined with your installation). Check the AXT documentation for more information on placing JAR files correctly.

Run the “`$AGETOR_HOME/data/axt/tracking/db/alfanumericgeneral.sql`” script to create the tables in the database. This file contains SQL commands to drop and create AXT Tracking database tables. Simply open the MySQL-client and run the script file with the MySQL “`\.`” command”. If your installation is placed in `c:\agetor` (on the windows platform) you execute the following “`\. c:/agetor/data/axt/tracking/db/alfanumericgeneral.sql`” from the MySQL client. This creates a new database with all needed tables.

If you have any problems installing or creating the database, please refer to the MySQL documentation.

 The above step deletes any data stored in the database. If you are upgrading an installation of version 2.x.x you should not perform this step. See section 2.2.3 instead.

The `general.sql` file contains standard SQL statements and thus can be used in any compliant relational database without modifications.

2.2.2 Important – converting data from AXT Tracking 1.0.x

If you are using AXT Tracking version 1.0.0, 1.0.1 or 1.0.2, it is important that you convert all your old references into the new database design. The old version did not use a relational database, but stored data as XML-files.

The conversion is done using a script, which parses the old references in the XML files, converts them and stores them in the new database. This script file is found in “`$AGETOR_HOME/data/axt/tracking/db/`” and are named `convert.bat` (convert on UNIX/Linux platform). This must be done *after* you’ve configured your database, created the database and the Tracking service is running without any errors.

Now run the `convert` program. If it gives any errors, please correct these in the xml files mentioned, run the database creating script and re-try the conversion.

After you have converted all the references, you should make a backup up your XML files found in your data-directory. After this is done clean the directory. This directory can now be used to save references that are deleted from the database (see section 2.1.3).

2.2.3 Important – updating the database from AXT Tracking 2.0.0

If you have an AXT Tracking 2.0 installation you will have to update the database before AXT Tracking 2.0.1 can be used. To update the database use the SQL script “\$AGETOR_HOME/data/axt/tracking/db/update.sql”.

2.2.4 Important – updating the database from AXT Tracking 2.0.2

If you have an existing installation of AXT Tracking you will have to update the database before AXT Tracking 2.0.3 can be used. To update the database use the SQL script “AGETOR_HOME/data/axt/tracking/db/convert_to_alphanumeric.sql”.

When using MySQL as the database you can open the mysql client and run the script with the “\c:\agetor\data\axt\tracking\db\ convert_to_alphanumeric.sql” command or use the MySQL Control Center.

3 Configuration

This section describes how to configure AXT Tracking. Each configuration file is described in the separate sections below.

3.1 Documents - tracking.xml

The “tracking.xml” file handles configuration of the incoming business documents. This file contains information about the setup of different kinds of documents, the properties you want to extract from these and how to determine the reference id tag.

The easiest way of understanding this file is by use of an example: A document and the corresponding “tracking.xml”, which uses the document.

A very simple order document (in XML format) could look like this:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xml>
  <orders>
    <order>
      <total>10</total>
      <articleid>102030</articleid>
    </order>
    <order>
      <total>5</total>
      <articleid>986745</articleid>
    </order>
  </orders>
</xml>
```

Figure 1 - XML Order

This document contains 2 orders each with an article id and how many articles wanted. The orders can be selected with the XPath¹ expression “/xml/orders/order”, thus this path is the document scope for each of the documents when extracting information in AXT Tracking. This means that there can be saved more than one document in AXT Tracking from a single XML document, each containing their own information relative to the document scope.

In this simple example there is no data within the document scope suitable to be the reference id. Thus AXT Tracking will create a new reference id and put the reference id in each of the root tags of the document scope “/xml/orders/order/axtrackref”. To add other (related) documents to this reference the returned reference id can be used.

¹ XPath is a XML standard for selecting parts of an XML-document. See “<http://www.w3.org/>” for more information.

The XML document doesn't tell which types of documents it contains. We have to define this in the configuration. Let us have a look on how the "tracking.xml" file could be configured to track the document.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<track>
  <confscope id='1'>
    <doc>
      <scopeid>/xml/orders/order</scopeid>
      <trackref></trackref>
      <type>Order</type>
      <prop>
        <name>total</name>
        <value>total</value>
      </prop>
      <prop>
        <name>articleid</name>
        <value>articleid</value>
      </prop>
    </doc>
  </confscope>
</track>
```

Figure 2 - "tracking.xml" order configuration

The file contains one or several confscope ID numbers "`<confscope>`", thus it is possible to have different document configurations in the "Tracking.xml" file. The confscope ID number is parsed from the filter configuration to make it possible to set up different filter documents in the "AXTConfiguration.xml". This is explained in section 3.3 in this document.

Each `<confscope>` contains one or more documents (identified by `<doc>`). This insures the possibility to have several different documents in the same file (for instance a file could contain both an order and an invoice). Each document has a scope id (`<scopeid>`), which is an XPath-expression pointing to the parts of the XML-document to be used.

Remember that in the order above, there wasn't any way to identify a reference number, so the tracking reference (`<trackref>`) is empty. Otherwise, you define where to find or place the reference id inside the confscope path, for example, "`<trackref>reference</trackref>`". This can be both a relative and an absolute XPath expression. An absolute XPath expression will start with "/" and contain the full path to the value.

The next tag is the type of the document (`<type>`), which can be an XPath to a variable in the XML document or a constant. In this example, it is a constant since no type definition exists in the document.

To extract data from the order we have to set up "*property extraction*". This is done using the property tag (`<prop>`). Inside each property tag is a tag denoting the name of the property (`<name>`). The property is saved using this name within AXT Tracking. There is also a value tag (`<value>`), which is an XPath-expression to the value of the property. In this case, the path is

relative to the scope id and must therefore be located within its scope. However, it is possible to locate a property anywhere in the XML-document if the value is an absolute Xpath-expression.

If you want to extract values from an attribute like the following:

```
<articleid no='10' />
```

You can simply use an XPath expression like the following:

```
articleid/@no
```

This set up will make two new references id's both containing an order document. Each of these order documents contains two properties, the "total" property and the "articleid" property. Two reference numbers are also put into the XML-document, since no reference id was there already.

Now, the XML-file looks like this:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xml>
  <orders>
    <order>
      <axttrackref>10001</axttrackref>
      <total>10</total>
      <articleid>102030</articleid>
    </order>
    <order>
      <axttrackref>10002</axttrackref>
      <total>5</total>
      <articleid>986745</articleid>
    </order>
  </orders>
</xml>
```

Figure 3 - XML Order processing result

The two reference numbers is proof of the document has been processed by AXT Tracking. This number is a unique key identifying the business process. Thus, when receiving a receipt with this number the system groups the order and receipt by the reference number.

Now, let's set up a receipt handling. First, take a look at this simple receipt:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<xml>
  <receipts>
    <receipt>
      <reference>
        <axttrackref>10001</axttrackref>
      </reference>
      <sender id='456kr7b' />
    </receipt>
  </receipts>
</xml>
```

Figure 4 - XML receipt

To handle this receipt, let us set up a new configuration scope in "tracking.xml". The document scope is "/xml/receipts/receipt", the reference number is placed under "reference/axttrackref" and we wish to save the sender id:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<track>
  <confscope id='1'>
    ...
  </confscope>
  <confscope id='2'>
    <doc>
      <scopeid>/xml/receipts/receipt</scopeid>
      <trackref>reference/axttrackref</trackref>
      <type>Receipt</type>
      <prop>
        <name>sender id</name>
        <value>sender/@id</value>
      </prop>
    </doc>
  </confscope>
</track>
```

Figure 5 - "tracking.xml" receipt configuration

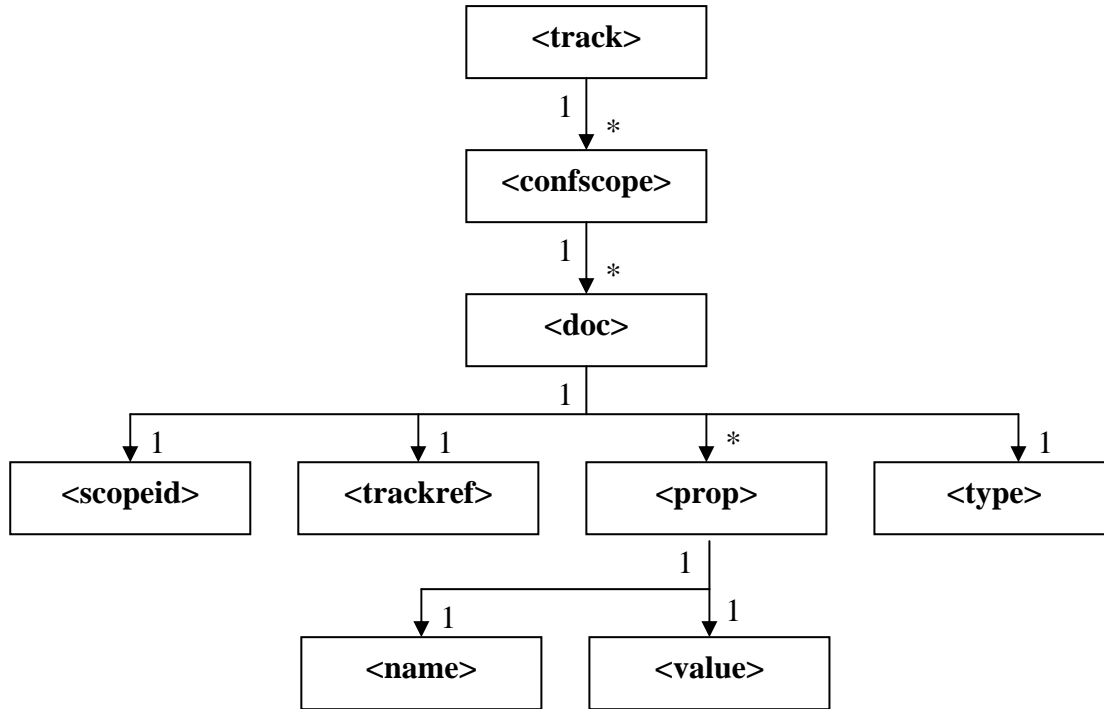
After receiving the above receipt, there are two reference id's in AXT Tracking, one with 2 documents (an order and a receipt 10001), and one reference only with the order 10002.

Let us have a look at a table showing the different tags used in "tracking.xml".

Tag	Optional	Description
<track>	No	This is the Root tag. In between this there can be several confscope tags.
<confscope>	No	This is a configuration. There can be more than one of these, if there

Tag	Optional	Description
		are several customers with different properties/settings. The confscope id is specified in the AXT filter configuration. In the <confscope> tag there is an attribute called “id”. This specifies the id of the configuration.
<doc>	No	In one configuration there can be more than one document scope. A <doc> tag is a document scope. This means that the document can be found once or several times in the stream. If the attribute “save” is set to “true”, the original document will be saved together with the properties and it can be showed in the AXT Tracking Administration Site. Default the document will not be saved.
<scopeid>	No	This is the scope id of the document scope and a made of an XPath expression. This specified where the scope exists in the stream.
<trackref>	Yes	With this you can specify if you want to locate the reference number tag elsewhere than in the root of the document scope. This XPath expression is relative to the document scope.
<type>	No	This tag specifies the type of the document scope. It can be done with a reference to the xml-stream, or just a constant name. I.E. order, invoice, receipt etc.
<prop>	No	This tag starts a property specification.
<name>	No	The name you wish the property to have.
<value>	No	An XPath expression where to locate the property and it is relative to the document scope.

The tags must be used beneath each other as illustrated below:



3.2 Events - tracking_event.xml

The configuration of the event handling reflects the dependencies between different documents in a business process. Let us use the same simple order/receipt example as in section 3.1.

An example of the “tracking_event.xml” could look like this:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<eventconf>
  <event doctype='Order'>
    <event doctype='Receipt' closes='true'>
      <warning time='1day'>
        <alarm media='mail'>
          <to>tracking@somecompany.com</to>
          <message>No receipt recieved </message>
        </alarm>
      </warning>
      <error time='1week'>
        <alarm media='sms'>
          <to>88776655</to>
          <message>No receipt recieved</message>
        </alarm>
      </error>
    </event>
  </event>
</eventconf>
```

Figure 6 - "tracking_event" order/receipt configuration

To reflect dependencies between documents the document types are placed in a *hierarchy*, which shows the order in which the documents enter AXT Tracking.

In this example we have two documents, an order and a receipt. First an order is received and afterwards a receipt is expected. It is possible to have several documents beneath one document, and the same document can occur several times in the set up.

Each event has a document type (doctype), which must reflect the document type used in the “tracking.xml” file. There is also an indicator whether or not this document is the one, which closes the reference id, meaning if this is the last of the documents in this flow of documents. In our example, we only have two documents, and the receipt is the last one, therefore it is a closing one. This is shown in the attribute “closes” in the <event> tag.

Each event can have a warning (<warning>) assigned and/or an error (<error>). These are triggered by the amount of time, between the arrivals of the document it self, and its parent document. In this example a warning is triggered after 1 day and an error is triggered after one week (<time>). This means, that if the receipt hasn’t arrived after 1 day, the reference goes into a warning state and further, if the receipt hasn’t arrived within a week, it goes into an error state. The state of a “reference id” is shown in the AXT Tracking site.

When an event (warning or error) is triggered, it is possible to perform an *action*. It is possible to send an email or an SMS to an administrator for him or her to take action. This is done by setting the media attribute in the alarm tag (<alarm>) to either email or sms and indicate which person this message (<message>) is to (<to>). If you wish to use these capabilities, it is important you configure the event properties correctly (see section 2.1.2).

If you only wish to set the state of the reference id and not send any notifications, just omit the media attribute from the alarm tag (<alarm>). Only define the message (<message>), which will be used on the site:

```

<alarm>
  <message>No receipt received</message>
</alarm>
```

Below is a table describing the tags used in the “tracking_event.xml” file and how they are used:

Tag	Optional	Description
<eventconf>	No	This is the root tag.
<event>	No	An event describes a document. There can be events under events, and that is the way you build up the flow of your documents. In this example, receipts are under the order. That indicates that the receipt is received after the order. It has two attributes: <ul style="list-style-type: none"> - “Doctype” (The type of the document) - “Closes” (Is this the last document coming in, and therefore the reference must be closed?)
<warning>	No	Is an event, which warns of a document missing. It has one attribute, which indicates the time limit (time), before the document must enter the system. It can be specified in the following: <ul style="list-style-type: none"> - second (I.e. “50second”) - minute (I.e. “40minute”) - hour (I.e. “30hour”) - day (I.e. “20day”) - week (I.e. “10week”)
<error>	No	Is an event, which warns of a document missing. It has one attribute, which indicates the time limit (time), before the document must enter the system. The attribute is the same as the warning above.
<alarm>	No	If you want an alarm in you warning/error, you can do it with this. It takes one attribute (media), which indicates which media you want to inform with. At the moment there are 2 types, “mail” & “sms”. Remember to set up your mail and sms information in the Tracking.properties file if you use this.
<to>	Yes	The to tag indicates which number, email etc. the alarm message is sent to. Optional if no media type is set.
<message>	Yes	The message, which is sent in the alarm, and/or shown in the AXT Tracking Administration Site.

3.3 Configuration – Filter set up

Setting up the AXT Tracking Filter in AXT is like setting up all other filters within AXT. This is done in the file “AXTConfiguration.xml”.

To set up a filter to hit the order document entry (confscopeid = 1) in the above “tracking.xml” file, do the following:

```
<!--
*****
      Order
*****
-->
<doc name="xmlorder">
  <key name="url" value="axttracking_order" />
  <filter class="dk.bording.axt.tracking.TrackingFilter">
    <param name="confscopeid" value="1" />
    <param name="encoding" value="ISO-8859-1" />
  </filter>
</doc>
```

Figure 7 - AXT configuration example

The filter has the name “dk.bording.axt.tracking.TrackingFilter”. The “confscope id”, which indicates which confscope id to hit in the **Tracking.xml**, and an **encoding** value reflecting which encoding the prolog must contain after the document has been processed by AXT Tracking.

 See the AXT Documentation for more on this topic.

3.4 Use of non XML-documents in AXT Tracking


It is possible to use text documents that are not XML in AXT Tracking without altering the configuration.

```
<!--
*****
      Non XML
*****
-->
<doc name="nonxml">
  <key name="url" value="axttracking_textorder" />
  <filter class="dk.bording.axt.tracking.TrackingFilter">
    <param name="confscopeid" value="1" />
    <param name="encoding" value="ISO-8859-1" />
    <param name="content" value="text"/>
    <param name="trackref" value="1001"/>
    <param name="total" value="10"/>
    <param name="articleid" value="1212"/>
  </filter>
</doc>
```

Figure 8 - Non XML content example

If you set the “content” parameter to “text”, AXT Tracking will not try to extract values from the document itself but instead use any the AXT parameters given to the filter. This means that it will use the content of the value in the configuration to search for an AXT parameter with the same name. In the configuration example showed previously in Figure 2 two attributes are defined as

“total” and “articleid” and there value can be set using parameters with the same name as in the above example. If the parameter “trackref” is used it will be used as reference id in AXT Tracking.

 The value attributes of the “param tags” can be filled out dynamically. How to do this is explained in the AXT Documentation.

3.4.1 Exceptions

The AXT Tracking filter throw exceptions, which is used in the AXT Filter routing system. You can route the document based on the outcome of running this filter. The table below summarizes the exceptions and their descriptions.

Exception	Description
TrackingMissingParameterException	This exception indicates a parameter is missing in the filter configuration. There are two required parameters. The conf scope Id, and the encoding. You must use both.
TrackingCfgException	This is thrown when a wrong path to the document type in the tracking.xml is used, or if the <axtrackref> path in the configuration doesn't match the document. Make sure to use correct paths.
TrackingInputException	If the incoming document doesn't match the conf scope id configuration. This is also thrown if a problem parsing the XML is encountered.
TrackingProcessException	If a problem processing the document in the filter arises. This could be due to communicating problems with the Service part of the AXT Tracking system or problems inserting the reference numbers.

To read more about the configuration and use of the basic AXT, filters, error routing etc., please refer to the AXT User Guide.

4 AXT Tracking administration site

To visualize the flow of documents, the event states, the saved information and the overall view of AXT Tracking, use the AXT Tracking Administration site. Point your browser to:

http://host_name/axt/tracking/

where the host_name is the DNS-name or IP-number of the machine where the installation is located.

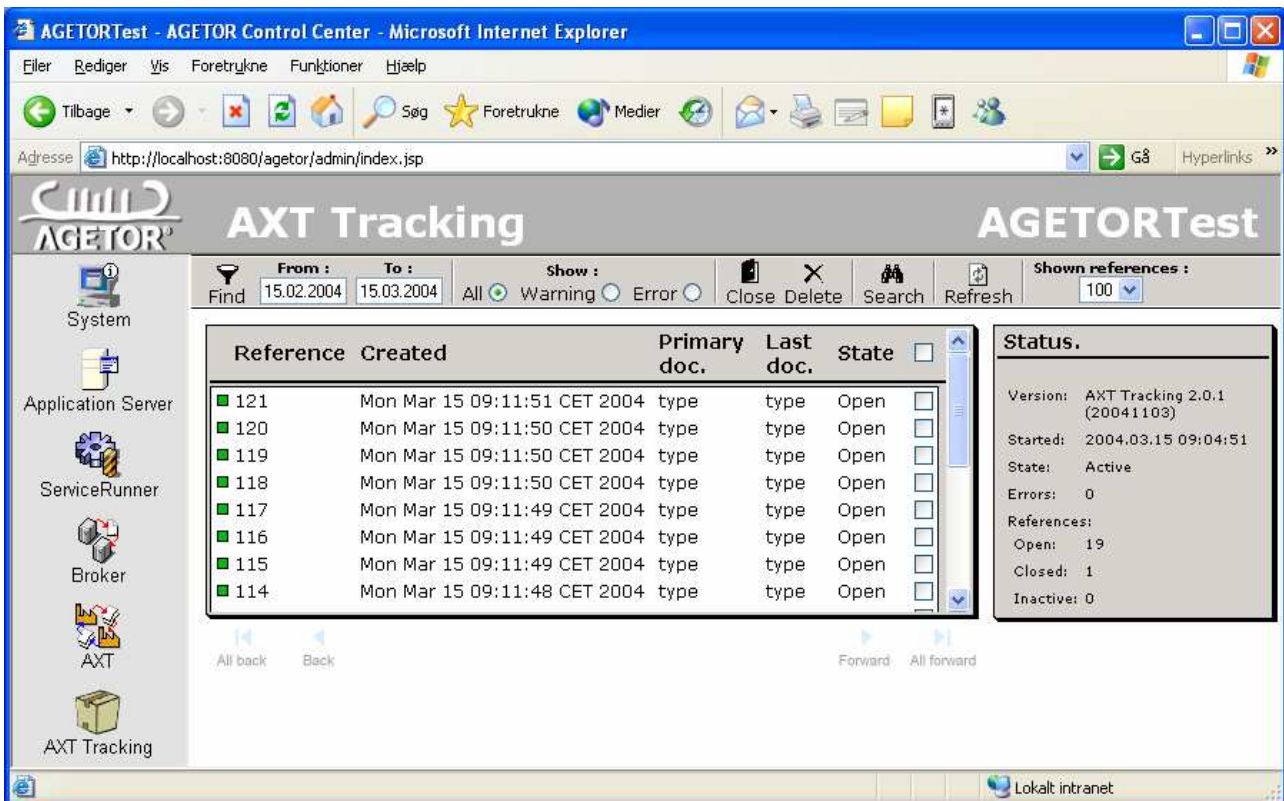
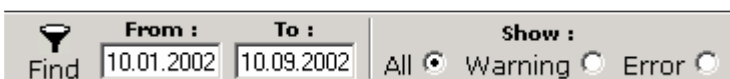


Figure 9 - The AXT Tracking Administration site

Let us examine this site.

4.1 Simple search

To do a simple search for all *open* references within a period of time, use the simple search:



You set the desired date interval. Both dates are included. To do an open-ended search, just fill in one of the date fields. For example, by only giving the date “to”, all references before this date is shown. Leaving out both date fields returns all open references.

There are 3 options available to you, defining which references are shown. To perform the search, click on the Find icon to the left.

- The *all* option is used to show all references, including errors and warnings.
- Choosing the *warning* option only shows the open references, which are in the warning state.
- Finally using the *error* option only shows the reference in error state.

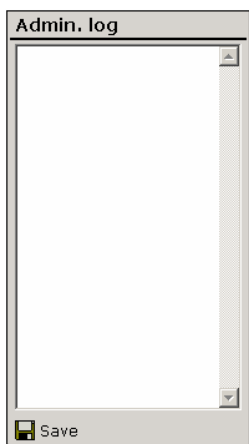
The references are listed in a main window. Let us have a look at them:

Reference	Created	Primary doc.	Last doc.	State	<input type="checkbox"/>
202	Mon May 27 00:00:00 CEST 2002	Order	Order	Warning	<input type="checkbox"/>
201	Mon May 27 00:00:00 CEST 2002	Order	Order	Warning	<input type="checkbox"/>
198	Mon May 27 00:00:00 CEST 2002	Order	Order	Warning	<input type="checkbox"/>

4.2 Sorting

Click on the columns headings to sort the list accordingly. Click once on any **heading** and the list is sorted in ascending order by that column. If the same heading is clicked again the list is sorted in descending order. This is possible with the reference and created headings.

4.3 Writing log to a reference id



It is possible to write text and attach it to the reference id. For instance, an order might be missing a receipt and is thus in an error state. You can therefore write in the log, that you have called the vendor about the missing receipt and it will be coming in a few days.

To write a log to a reference, simply click on the reference id to zoom into it. Now you can write your text in the box to the right of the screen. After you have written your text, click the save button at the bottom of this window to save your log.

4.4 Reference detail

To see all the documents to a reference id, their properties, state etc, simply click on the reference id number. This opens the reference:

doctype	timestamp	Afsender	Modtager
Order	Mon May 27 15:54:11 CEST 2002	5790001060428	5790000712212

no invoice on order

All documents and the properties extracted from these are listed. If you click on the reference again it closes.

If the document has been saved an icon appears in the list of documents under a reference.



When the document icon is clicked it opens a new window in the browser showing the original document.

4.5 Closing / Deleting references



You can close or delete one or more references, choosing them by clicking in the select box, which are in the right side of the list. You can select one or more, or all by clicking at the select box in the top of the list. After you have chosen them, simply just click on the close/delete icon in the toolbar.

Remember, that deleting a reference puts the reference in inactive state. It's only deleted after the time period you have specified in your properties (refer to section 2.1.3).

4.6 Advanced Search

With the advanced search function it is possible to search for specific references containing specific document types with specific properties. Click the search icon in the toolbar. This opens the advanced search input fields as shown:


Document Search

Search by reference ID:
ID:

Or search after properties:
In doc. type:

Name	Value
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Incl. closed references.

 Search

There are three ways to search: a specific reference id, by document type or by properties. You can combine the last two, thus searching for references with specific properties.

To search on a specific reference id you type the name into the ID field and press the search icon showed in the bottom to the right.

Search using wildcards are supported in SQL syntax:

_ is a single character wildcard.

* is a string wildcard.

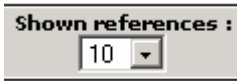
ie. "*GEN*" will find all ids containing the string GEN.

To search for references with a specific document type, put this type in the doc. type value text field.

To search for properties, put them into name value pairs in the text fields below. If you also have specified a document type, only properties in this document type are searched. If you select incl. Closed references also closed references are returned.

4.7 Choose the number of shown reference

You can choose how many references to show on each page. The default number is 100, but you can change this to suit your taste. This is done from the toolbar where a dropdown box is found.



4.8 Current Status

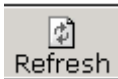
When you not are zoomed into a references detail a status is displayed to the right.



The status contains basic information on how AXT Tracking is doing. It contains the current version, the time the server was started and which state the server is in. The status also contains the number of errors occurred since the server was started. Finally, the status shows how many references the tracking system contains. These are divided into open, closed and inactive references. An in-active reference is a reference ready to be deleted from the system. Read more about this in the delete configuration set up in the Installation guide.

4.9 Refreshing

You can refresh and thereby update the reference list by pushing the refresh icon in the toolbar.



5 Configuration problems

This section tries to address common problems and give solutions to these problems.

“No document entries found for any document scope in the input stream with the ...”

- The error arises when the input stream doesn't match the configuration in the **tracking.xml**. The AXT Tracking didn't find any documents inside the stream with the document scope you have used in the **tracking.xml**.
- You must ensure the XML file, which is send to the filter, does contain one ore more of the documents indicated in the conf scope within **tracking.xml**. The document scope must be an Xpath which exist in the XML file. Make sure the path is correct!

“There was a problem inserting the reference number into the xml-document ...”

- If you are using the <trackref> tag in the **tracking.xml** file, indicated path doesn't exist in the incoming XML-file. Make sure this path is correct. Remember this tag is relative to the document scope and must be beneath it.

“The <type> tag which is used in the Tracking Configuration file ...”

- If you have used the <type> tag in the **tracking.xml** file this error informs you that path used is wrong. Make sure the path is correct and correspond to the incoming XML-file. Remember this path is **not** relative to the document scope, and can point anywhere in the document.

“Could not communicate with service: ...”

- Make sure the AXT Tracking services and the service runner are running.

“No matching document found for ...”

- This indicates the keys given don't reflect the configuration of the document entry in the **AXTConfiguration.xml** file.
- Make sure the input keys match the one in the configuration file.

“The root element is required in a well-formed document...”

- Make sure input xml-document is a well-formed one.

“Error (1000): Can't connect to database. (Error: ERR_SERVICECONTACT)”

- Make sure the AXT Tracking services and the services runner are running. Perhaps take a look in the AXT Tracking Installation guide.