Adabas Analytics on Linux, UNIX and Windows

Version 2.1 - Innovation Release

April 2017

This document applies to Adabas Analytics on Linux, UNIX and Windows Version 2.1 - Innovation Release and all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: EAL-DOC-21-20170411

Table of Contents

Preface

This documentation describes the product Adabas Analytics for Linux, Unix and Windows platforms.

1 Concepts

Typically, an Adabas database is used in a commercial environment, and the data contained in the database are usually of a sensitive and confidential nature. Seen in this context, it is important to be able to answer the following questions (sometimes called the 5 W questions):

- Who has accessed the data?
- What has been accessed? This includes the database ID, the file number, the type of access (create, read, update, delete), the field names, etc.
- When was the data accessed?
- Where was the data accessed from?
- What has changed in the internal state of the database?

These 5 questions are of vital importance for the following reasons:

Fraud prevention
Identify security incidents in operational databases; who is accessing sensitive data?

Auditing
Keep track of and analyse compliance-relevant results; who did what, from where and when?

Performance monitoring
Central diagnosis of database performance and efficiency; how well is Adabas running?

Adabas Analytics adresses these requirement by enabling you to create an event each time there is a change of state in the Adabas nucleus.

A change of state can be triggered by:

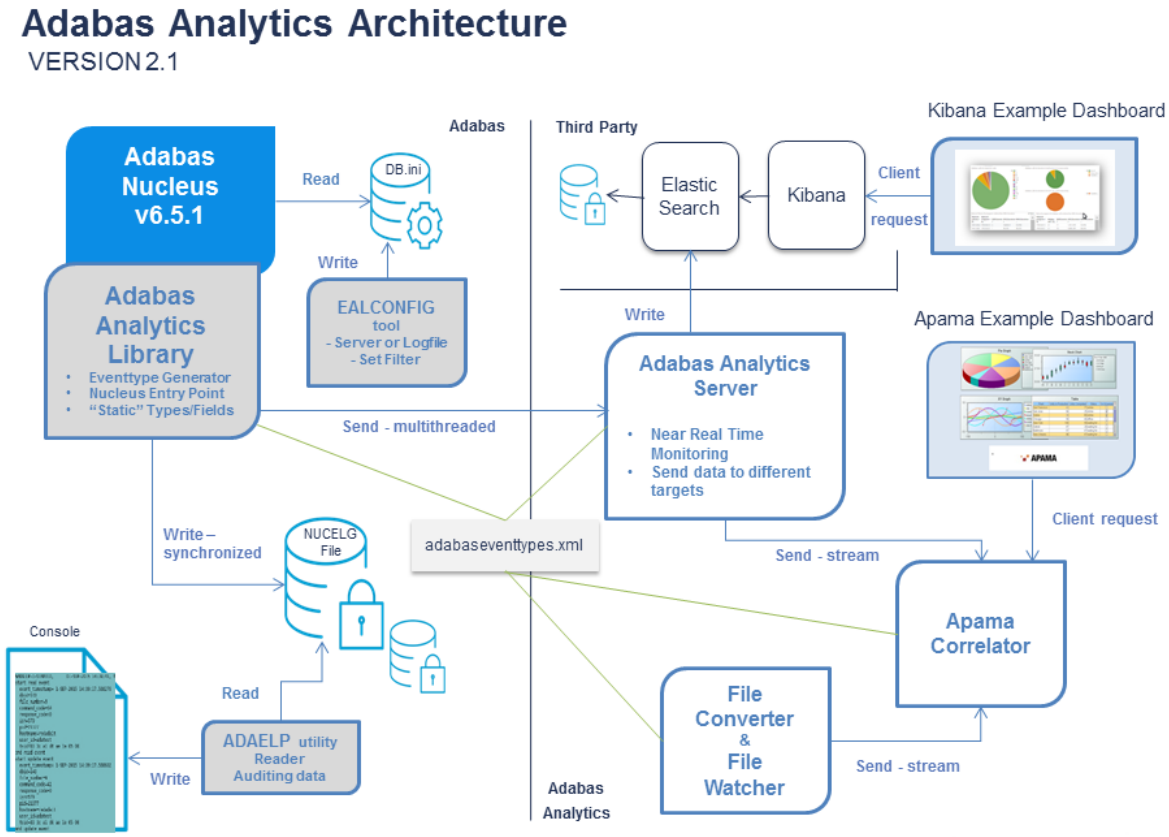
- An Adabas call;
- A security event (authorization succeeded or failed, authentication succeeded or failed, etc.);
- A change in performance status (threshold reached, disk space exhausted, etc.).

Adabas Analytics currently supports 14 types of events related to Adabas calls. For further information about the event types, see the section ***Adabas Analytics Event Types***. More event types relating to security and performance will be supported in later versions.

Because you might only need to use Adabas Analytics sporadically (depending on your use case), it is possible to activate/deactivate the event anayltics component. Also, because only certain event types might be of interest, you can easily filter events by file number and event type.

The events generated by Adabas Analytics are written to a local log file called NUGELG; you can display the contents of this file with the new Adabas utility ADAELP (for further information, see the section ***ADAELP (Event Log Report)***.

The following graphics shows the architecture of Adabas Analytics Version 2.1:



Adabas Analytics and Elasticsearch/Kibana

Starting with Version 2.1, Adabas Analytics uses Elasticsearch and its visualization component Kibana to store and visualize Adabas performance data. The combination of the Adabas nucleus, Adabas Analytics, Elasticsearch and Kibana lets you analyze Adabas performance data in near-realtime. The Kibana visualization of the data is in a browser interface.

The installation kit includes an example Kibana dashboard, which you can use to display Adabas performance data.

Adabas Analytics and Apama

The Adabas nucleus creates the Adabas Event Logfiles (`NUCELG.xxxx`) if the Adabas Eventing functionality is enabled.

The Adabas Analytics File Converter reads a single Adabas Event Logfile and sends it to the Apama Correlator.

In the Apama Correlator, the received events can be processed like any Apama event: use them in an Apama monitor or an Apama Correlator dashboard.

The Adabas Analytics File Converter and the Apama Correlator can be running on the same node or on distributed nodes.

2 Installing Adabas Analytics

The Adabas Analytics is installed using the Software AG Installer. Please refer to *Using the Software AG Installer* for detailled information about how to use the installer.

Prerequisites

The following prerequisites must be met for this version of Adabas Analytics:

- Java Version 1.8 or higher; an appropriate Java runtime is provided during the installation.
- Adabas Version 6.5.1 or higher.
- Natural Version 8.4.1 or higher (only required if new event types like ADA_NAT_PERF are to be monitored). This version is only available on request.

Supported Operating System Platforms (Linux/Unix)

Adabas Analytics supports the following operating system platforms:

- AIX 7.1 (Power 64 bit)
- AIX 7.2 (Power 64 bit)
- HP-UX 11.i v3 (Itanium 64bit)
- Red Hat Enterprise Linux Server 6 (IBM System z 64bit)
- Red Hat Enterprise Linux Server 7 (IBM System z 64bit)
- Red Hat Enterprise Linux Server 6 (x86-64)
- Red Hat Enterprise Linux Server 7 (x86-64)
- Oracle Solaris 11 (SPARC 64bit)
- SUSE Linux Enterprise Server 11 (IBM System z 64bit)
- SUSE Linux Enterprise Server 11 (x86-64)
- SUSE Linux Enterprise Server 12 (x86-64)

Supported Operating Platforms (Windows)

Adabas Analytics supports the following operating system platforms:

- Windows Server 2008 R2 (Standard and Enterprise Edition, x86-64)
- Windows Server 2012 (Standard and Datacenter Edition, x86-64)
- Windows Server 2012 R2 (Standard and Datacenter Edition, x86-64)
- Windows 7 (Professional, Ultimate and Enterprise Edition, x86-64)
- Windows 8 (Pro and Enterprise Edition, x86-64)
- Windows 10 (Pro and Enterprise Edition, x86-64)

Home Editions of Microsoft Windows are not supported.

Notes:

1. We suggest that you install all of the recommended and important Microsoft Windows updates before you start the installation.
2. On Windows 8.1 and Windows Server 2012 R2 the installation will fail if the Microsoft update KB2919355 is missing.

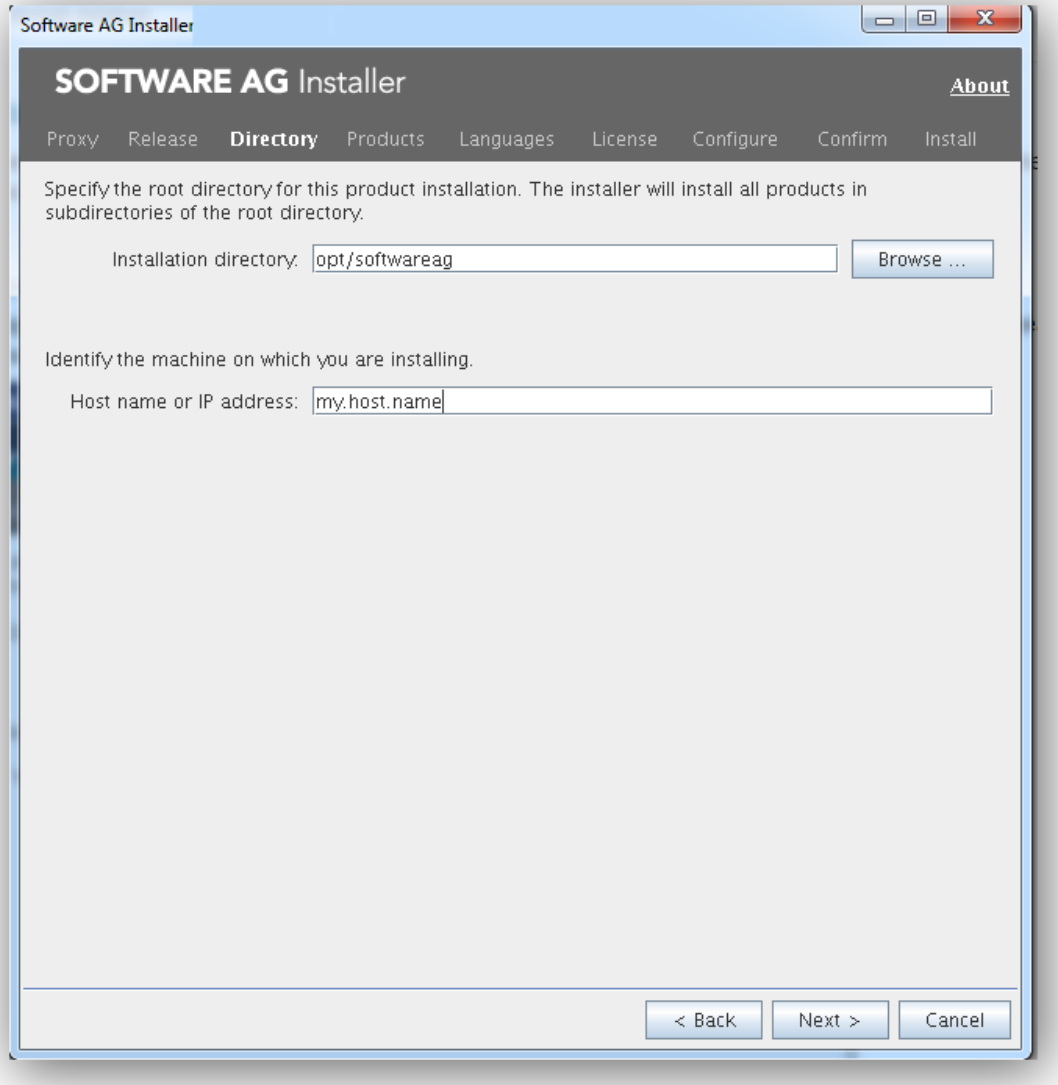
Installing on Linux/Unix

This installation documentation provides just a brief description on how to install Adabas Analytics directly on the target machine using the Software AG Installer GUI. For detailed information on the Software AG Installer, see *Using the Software AG Installer*.

To install Adabas Analytics

1 Start the Software AG Installer GUI as described in *Using the Software AG Installer*.

2 When the first page of the Software AG Installer GUI (the so-called Welcome panel) is shown, press the **Next** button repeatedly (and specify all required information on the shown panels as described in *Using the Software AG Installer*) until the panel containing the installation directory appears.



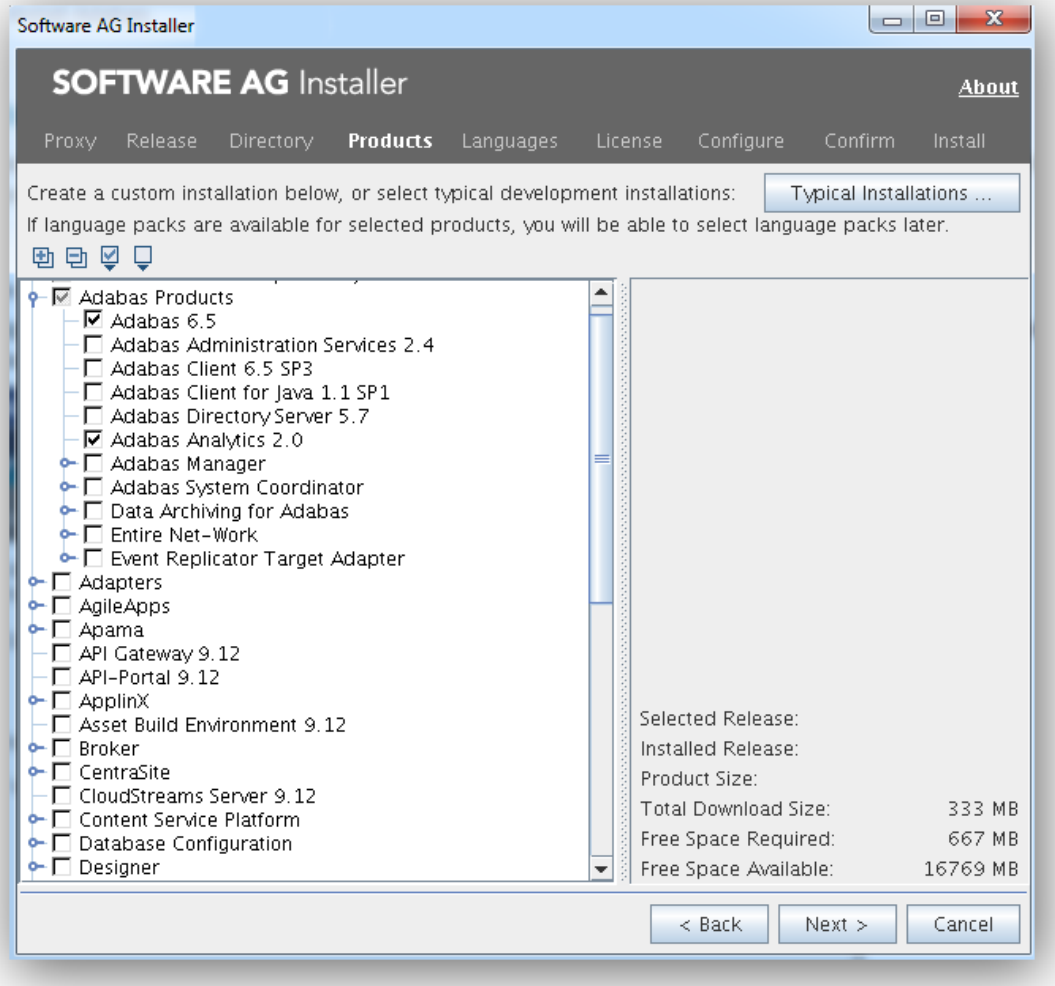
3 Specify the root directory and host name or IP address (optional).

Note: It is strongly recommended not to use the proposed default directory */opt/softwareag* but a sub-directory, for example */opt/softwareag/adabasanalytics21* to install the release version of Adabas Analytics 2.1. With this approach you can install several versions of Adabas Analytics in parallel directories.

4 Press the **Next** button.

Note: The panel shown below is an example of a possible product selection.

The panel containing the product selection tree appears. This tree lists the products for which you have valid credentials and which can be installed on the operating system of the machine on which you are installing.



Note: Products or product versions which are already installed in the selected installation directory are shown as disabled.

5 If you want to install Adabas and pre-selected product components, select the **Adabas Products** node.

Or:

If you want to customize the list of selected product components, expand the **Adabas Products** node, deselect Adabas Products and select the product components that you want to install.

6 If you want to install Adabas Client, select **Adabas Client** in the product selection tree. The Adabas Client is always installed together with Adabas, but can also be installed separately.

7 Press the **Next** button.

8 Read the license agreement, select the check box to agree to the terms of the license agreement, and press the **Next** button.

9 Specify whether to use sudo or not.

Some parts of the installation require root permissions. On the following sudo panel you must either select **Use sudo, with password** suppling a valid sudo password or you can skip these installation steps by selecting **Do not use sudo or sudo is not available**.

You will then have to execute those steps as described on the panel shown below. Both alternatives are equivalent.

Note: Using sudo without specifying a password is not possible.

10 On the last panel, review the list of products and items you have selected for installation. If the list is correct, press the **Next** button to start the installation process.

When the Software AG Installer has completed the first-time installation, additional configuration steps are required. See *Configuring Adabas Anlytics* for further details.

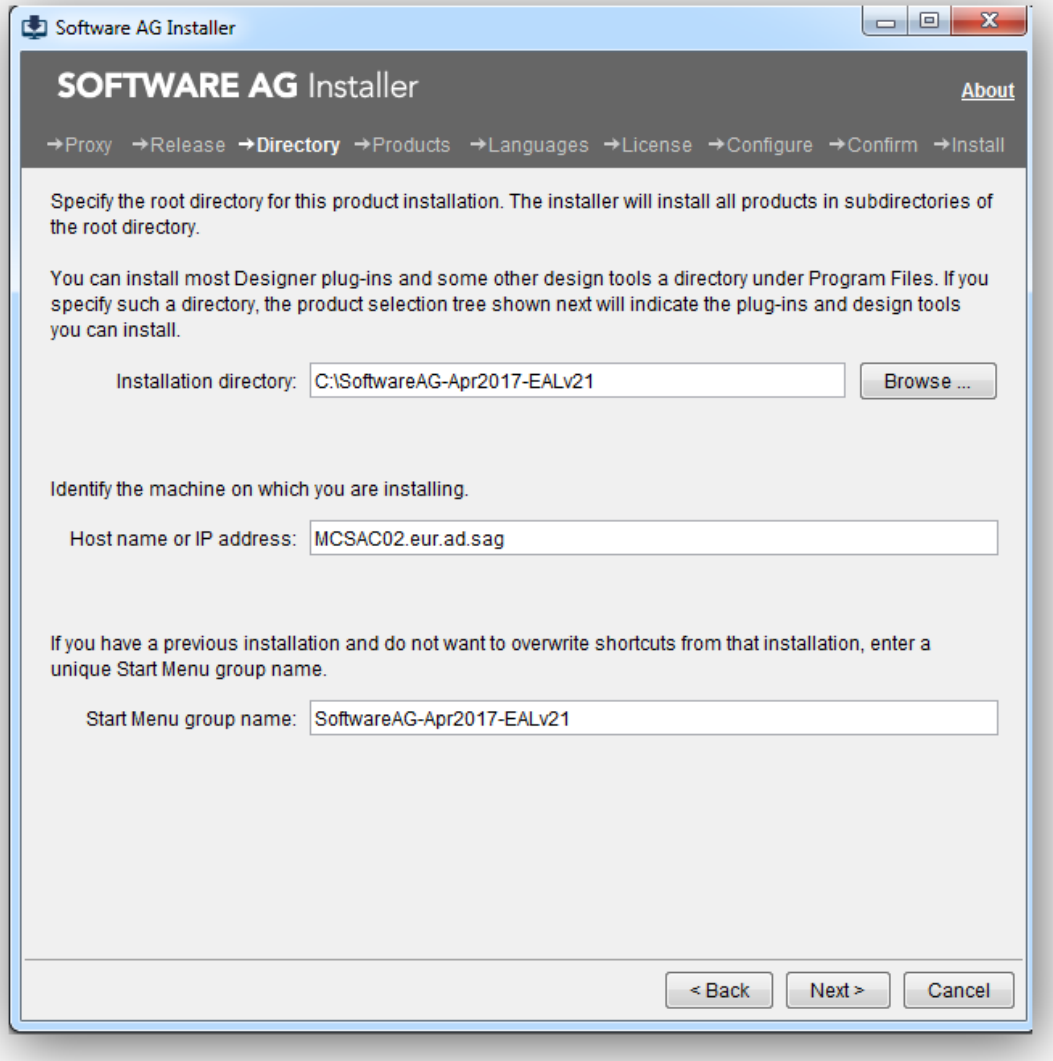
Installing on Windows

This installation documentation provides just a brief description on how to install Adabas Analytics directly on the target machine using the Software AG Installer GUI. For detailed information on the Software AG Installer, see *Using the Software AG Installer*.

To install Adabas

1 Start the Software AG Installer GUI as described in *Using the Software AG Installer*.

2 When the first page of the Software AG Installer GUI (the so-called Welcome panel) is shown, press the **Next** button repeatedly (and specify all required information on the shown panels as described in *Using the Software AG Installer*) until the panel containing the installation directory appears.



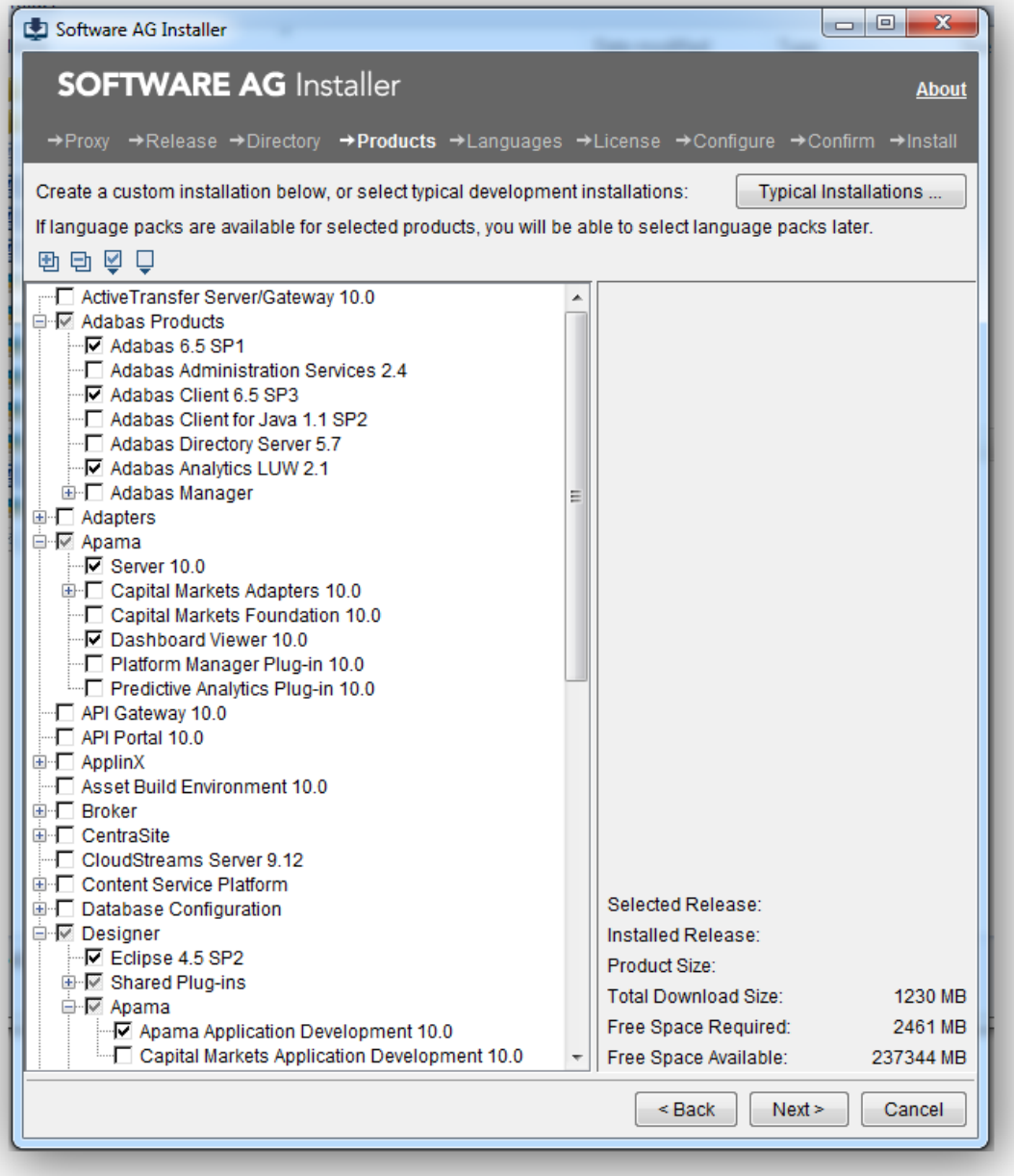
3 Specify the installation directory, host name or IP address (optional) and the Start Menu group name.

Note: It is strongly recommended not to use the proposed installation directory *C:\SoftwareAG* but a sub-directory, for example *C:\SoftwareAG\AdabasAnalytics21* and to change the Start Menu group name, for example to **Software AG Adabas Analytics 2.1** to install the release version of Adabas Analytics 2.1. With this approach you can install several versions of Adabas Analytics in parallel directories.

4 Press the **Next** button.

Note: The panel shown below is an example of a possible product selection.

The panel containing the product selection tree appears. This tree lists the products for which you have valid credentials and which can be installed on the operating system of the machine on which you are installing.



Note: Products or product versions which are already installed in the selected installation directory are shown as disabled.

5 If you want to install Adabas and pre-selected product components, select the **Adabas Products** node.

Or:

If you want to customize the list of selected product components, expand the **Adabas Products** node, deselect Adabas Products and select the product components that you want to install.

6 Press the **Next** button.

7 Read the license agreement, select the check box to agree to the terms of the license agreement, and press the **Next** button.

8 On the last panel, review the list of products and items you have selected for installation. If the list is correct, press the **Next** button to start the installation process.

When the Software AG Installer has completed the first-time installation, additional configuration steps are required. See *Configuring Adabas Analytics* for further details.

3 Getting Started

Once you have successfully installed Adabas Analytics, there are some steps that have to be performed before you can start to collect and display Adabas events:

1. Use the configuration tool EALCONFIG to make the entries in the *DBnnn.INI* file that are required to enable collecting Adabas events. Pleaser refer to *EALCONFIG (Event Analytics Configuration Tool)* for further information.
2. Use the event file converter to read and convert the contents of an existing Adabas event file, prior to displaying them in an Apama dashboard. Please refer to *The Event File Converter* for further information.
3. Install the third-party products Elasticsearch and Kibana if you want to visualize Adabas data in Kibana. Start the Adabas Analytics server to allow near real-time monitoring of your Adabas/Natural application. Please refer to *Getting Started with Elasticsearch/Kibana* for further information.

4 Release Notes

This chapter gives an overview of the features of Adabas Analytics Version 2.1 that have been introduced or modified since the previous release (Version 2.0).

The chapter contains the following sections:

General Information

This section provides information which you should be aware of before you install and use Adabas Analytics Version 2.1.

Supported Operating Systems

Adabas Analytics supports the same operating systems and patforms as Adabas for Linux, UNIX and Windows 6.5.1.

Software AG Installer

Adabas is now installed using the Software AG installer. Please refer to the relevant installation documentation for further information.

New, Modified and Dropped Features in Version 2.1

Force Switch of NUCELG File (New)

You can now force a switch of the NUCELG file using the new ADAOPR function FEOF=ELOG.

Adabas Analytics Server (New)

The Adabas Analytics server, which is new with Version 2.1, helps to avoid bottlenecks when writing events to the NUCELG file, and is a first step towards near real-time monitoring. The Adabas Analytics server can be configured to support different targets - the Apama Correlator, or the new, third-party product Elasticsearch.

Note: It is still possible to send NUCELG files to Apama using the Adabas Analytics File Converter.

Elasticsearch/Kibana (New)

This version is delivered with the third-party products Elasticsearch and Kibana, which can be used to visualise data from the Adabas Analytics server in near real-time. An example Kibana dashboard is also provided.

EALCONFIG (Modified)

Compared to the version provided with the Adabas Version 6.5.0 package, the new version of EALCONFIG has the following major differences:

- New subtopics TARGET_EAL_SERVER and TARGET_NUCELG;
- The items SWITCH_AFTER_EVENTS and SWITCH_AFTER_TIME are moved to TARGET_NUCELG;
- The item TARGET_NUCELG is ignored if TARGET_EAL_SERVER exists;
- The new event types NAT_INSERT, NAT_READ, NAT_UPDATE, NAT_DELETE, NAT_COMMIT, NAT_ROLLBACK, ADA_PERF and ADA_NAT_PERF are supported.

New, Modified and Dropped Features in Version 2.0

Adabas Extensions for Adabas Analytics

You can configure the database INI files for use with Adabas Analytics with the tool EALCONFIG. You can use the utility ADAELP to print events from an event log created by Adabas. Both of these components are part of the Adabas kit.

Adabas Analytics Version 2.0 no longer uses the Adabas replication exit; all of the functions required to trigger collection of event data to the NUCELG file are now part of the Adabas kernel Version 6.5 and above.

Event File Converter

The Event File Converter is a program that sends a NUCELG file as a stream to an Apama correlator. Other targets are currently not supported.

Apama Dashboard

This version provides an example Apama dashboard, which can be imported into the Eclipse-based Software AG Designer.

Documentation Updates and Changes

The most recent product documentation, hotfixes and other useful information can be found in Empower.

5 The Event File Converter

The Adabas nucleus creates the Adabas event log files NUCELG.xxxx if the Adabas Eventing functionality is enabled. The Adabas Analytics event file converter reads an Adabas event log file and sends it to the Apama Correlator, where the events can be processed like any other Apama event, and displayed in an Apama monitor or in an Apama Correlator dashboard.

Note: The Adabas Analytics file converter and the Apama Correlator can be running on the same node or on distributed nodes.

Using the Event File Converter

Starting the Event File Converter on Windows

The Windows start menu contains an entry for a command prompt, from where you can call the event file converter. In the command prompt window, issue the command:

```
AdabasAnalyticsFileConverter.bat ↩
```

Starting the Event File Converter on UNIX

Before you can start the event file converter, you must first source the environment file *ealenv*. Alternatively, you can source the top-level environment file *sagenv.new*, which in turn sources *ealenv*. Then issue the command

Usage of the Event File Converter

The event file converter is a command line utility with this syntax:

```
AdabasAnalyticsFileConverter -f <name> -t <host[:port]> [-h]
```

where:

-f, eventfile <name>
The name of the event file to be read.

-t, target <host[:port]>
Send the events from the event file to the Apama Correlator; the default port number is 15903.

6 Apama Example Dashboard

You can use the example Apama dashboard provided with the installation to display the events contained in an event log file.

The following files and folders are provided with the installation:

File/Folder	Description
AdabasEvents.mon	The event definitions file for Adabas events.
ApamaExampleDashboard	Adabas Analytics example Apama dashboard application, this has to be imported into the Eclipse-based Software AG Designer.

Note: the dashboard provided is just an example. Please refer to the Apama documentation for details about how to build your own dashboard.

Using the Software AG Designer

Before you can use the example provided, or develop your own applications in Apama, the event definitions for Adabas events have to be imported into an Apama project. In the Software AG designer, import the *AdabasEvents.mon* file into subdirectory *eventdefinitions*.

The file *AdabasEvents.mon* is also a part of the Apama dashboard example (*ApamaExampleDashboard* folder), which can be imported using the Software AG Import wizard.

To run the example as an Apama project in the Software AG Designer

1 Open the Software AG Designer.

2 Select **Import** from the **File** menu.

3 In the Import wizard, select and expand the *General* node, then select **Existing Projects into Workspace**.

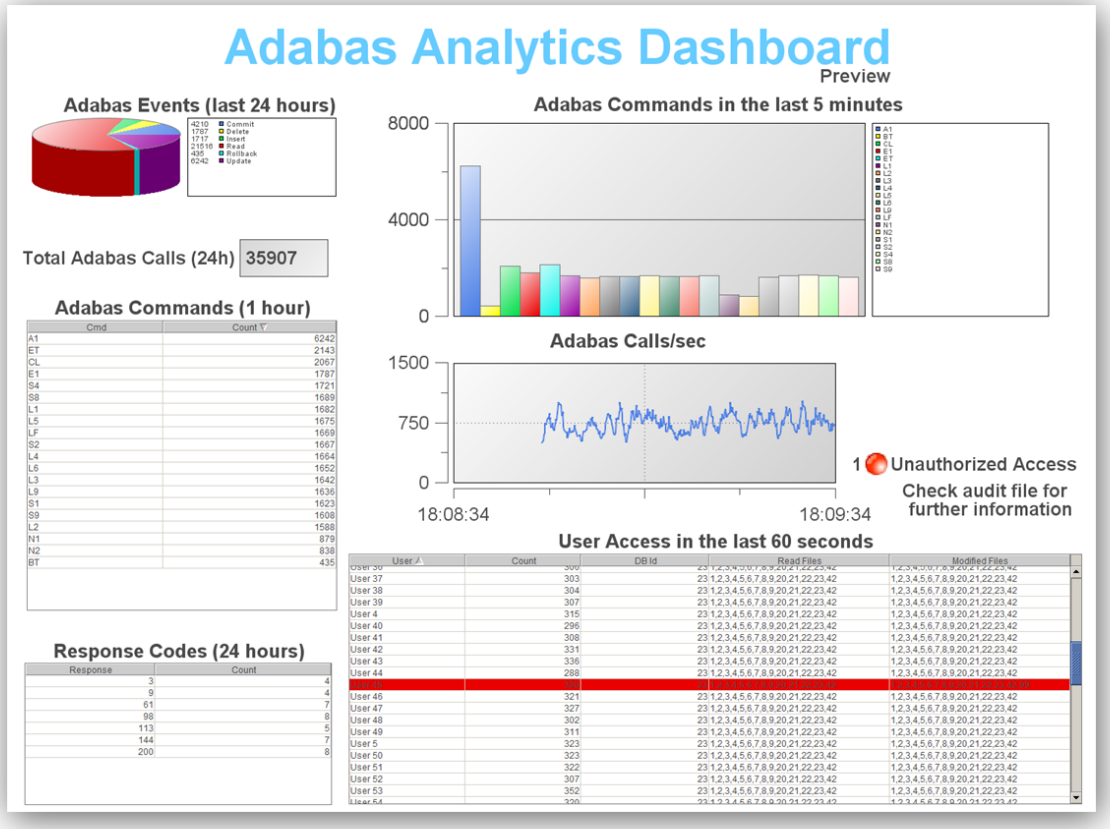
4 Click the **Next** button, and the click on the **Browse** button in the Import Project step.

5 Navigate to *<installation directory>\AdabasAnalytics\apama\ApamaExampleDashboard* and select that folder.

6 In the **Options** panel of the **Import Projects** dialog, check the **Copy projects into workspace** check box.

7 To run the example, right-click the project and select **Run As** -> **Apama Application** from the **Apama Developer Perspective**. Then click on the **Start** button in the **Launch Control Panel** of the Apama Workbench Perspective.

The following shows an example of how the dashboard might look:


Adabas Analytics Dashboard
Preview
Adabas Events (last 24 hours)
4210 Commit
1787 Delete
1717 Insert
21516 Read
435 Rollback
6242 Update
Total Adabas Calls (24h) 35907
Adabas Commands (1 hour)
Cmd Count
A1 6242
ET 2143
CL 2067
E1 1787
S4 1721
S8 1689
L1 1682
L5 1675
LF 1669
S2 1667
L4 1664
L6 1652
L3 1642
L9 1636
S1 1623
S9 1608
L2 1588
N1 879
N2 838
BT 435
Response Codes (24 hours)
Response Count
3 4
9 4
61 7
98 8
113 5
144 7
200 8
Adabas Commands in the last 5 minutes
8000
4000
0
Adabas Calls/sec
1500
750
0
18:08:34
18:09:34
1 Unauthorized Access
Check audit file for further information
User Access in the last 60 seconds
User Count DB Id Read Files Modified Files
User 37 303 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 38 304 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 39 307 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 4 315 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 40 296 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 41 308 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 42 331 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 43 336 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 44 288 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 45 1 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 46 321 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 47 327 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 48 302 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 49 311 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 5 323 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 50 323 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 51 322 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 52 307 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42
User 53 352 23 1,2,3,4,5,6,7,8,9,20,21,22,23,42 1,2,3,4,5,6,7,8,9,20,21,22,23,42

7 Elasticsearch/Kibana

Adabas Analytics can be used to analyze the performance of an Adabas database; it is possible to examine the performance data of each Adabas call executed in the database.

The current version of Adabas Analytics is delivered with the third-party product **Elasticsearch** (and its visualization component **Kibana**) to store and visualize the Adabas performance data. The combination of the Adabas nucleus, the Adabas Analytics server, Elasticsearch and Kibana lets you analyze Adabas performance data in near-realtime.

Getting Started with Elasticsearch/Kibana

Prerequisites

Adabas Version 6.5.1 and Adabas Analytics Version 2.1.0 must already be installed before you install and use Elasticsearch/Kibana.

The following default TCP ports are used:

- 6521, 6522 - Adabas Analytics server
- 9200, 9300 - Elasticsearch
- 5601 - Kibana

Please ensure that these ports are not already in use. Depending on your configuration and/or firewall settings, open this ports for inbound communications.

Installing, Configuring and Starting Elasticsearch

To install, configure and start Elasticsearch

1 Install the third-party software Elasticsearch from *$EALPROGDIR/third-party* by extracting the file *elasticsearch-5.1.2.zip* to a directory <*ELASTICSEARCH_INSTALL_DIR*> of your choice (Windows), or by extracting the file *elasticsearch-5.1.2.tar* to a directory <*ELASTICSEARCH_INSTALL_DIR*> of your choice (Unix).

2 Edit the Elasticsearch configuration file: the configuration file *elasticsearch.yml* is located in the *config* subdirectory of <*ELASTICSEARCH_INSTALL_DIR*>.

Change the line

```
#cluster.name: my-application
```

to

```
cluster.name: AdabasAnalyticsData
```

If the Adabas Analytics server and Elasticsearch are running on different hosts, change the line:

```
#network.host: 192.168.0.1
```

to

```
network.host: 0.0.0.0
```

Otherwise Elasticsearch will not be able to connect with the remote Adabas Analytics server.

You can find detailed information about configuring Elasticsearch here: **https://www.elastic.co/guide/en/elasticsearch/reference/current/settings.html**.

3 Start Elasticsearch by executing *ELASTICSEARCH_INSTALL_DIR\bin\elasticsearch.bat* (Windows), or *ELASTICSEARCH_INSTALL_DIR/bin/elasticsearch* (UNIX).

Elasticsearch should now be up and running on localhost:9200.

4 If Elasticsearch is running on a different host to the Adabas Analytics server, edit the Adabas Analytics configuration file *EALPROGDIR/configuration/config.xml*.

Change the line

```
<Location host="localhost" port="9300" />
```

to

```
<Location host="<ELASTICSEARCH_HOST>" port="9300" />
```

where <ELASTICSEARCH_HOST> is the name of the host on which Elasticsearch is running.

Installing, Configuring and Starting Kibana

To install, configure and start Kibana

1 Install the third-party software Kibana from *$EALPROGDIR/third-party* by extracting the file *kibana-5.1.2-windows-x86.zip* a to a directory <*KIBANA_INSTALL_DIR*> of your choice (Windows), or by extracting the file *kibana-5.1.2-linux-x86_64.tar.gz* to a directory <*KIBANA_INSTALL_DIR*> of your choice (UNIX).

2 Edit the Kibana configuration file: the configuration file *kibana.yml* is located in the *config* subdirectory of <*KIBANA_INSTALL_DIR*>.

Change the line

```
#elasticsearch.ssl.verify: true
```

to

```
elasticsearch.ssl.verify: false
```

If Kibana is installed on a different host to Elasticsearch, change the following lines (leave them unchanged if Kibana is installed on the same host as Elasticsearch):

Change

```
#server.host: "localhost"
```

to

```
server.host: <KIBANA_HOSTNAME>
```

where <KIBANA_HOSTNAME> is the name of the host on which Kibana is installed.

Change

```
#elasticsearch.url: "http://localhost:9200"
```

to

```
elasticsearch.url: "http://<HOSTNAME>:9200"
```

where <HOSTNAME> is the name of the host on which Elasticsearch is installed.

Change

```
#server.name: "your-hostname"
```

to

```
server.name: "your name"
```

You can find detailed information about configuring Kibana here: **https://www.elastic.co/guide/en/kibana/current/settings.html**.

3 Start Kibana by executing *KIBANA_INSTALL_DIR\bin\kibana.bat* (Windows), or *KIBANA_INSTALL_DIR/bin/kibana* (UNIX).

Kibana should now be up and running on localhost:5601.

Visualizing Adabas Analytics Data using Elasticsearch/Kibana

You can use the example Kibana dashboard provided with the installation to display Adabas/Natural performance data in near-realtime. The following section describes how to import and use the dashboard.

To visualize Adabas Analytics data using Elasticsearch/Kibana

1 Start the Adabas Analytics server.

On Windows:

```
EALPROGDIR/bin/AdabasAnalyticsServer.bat start
```

On UNIX:

```
EALPROGDIR/bin/AdabasAnalyticsServer.sh start
```

2 Create a demo Adabas database with the *crdemodb* utility.

```
crdemodb <dbid>
```

3 Run the configuration utility *EALCONFIG*, which is located in the directory *ADAPROGDIR*. When you are prompted for filter events, select ADA_NAT_PERF from the list of available event types.

For further information about the *EALCONFIG* utility, see the section ***EALCONFIG (Event Analytics Configuration Tool)***.

4 Start the demo Adabas database that you created earlier.

```
adastart <dbid>
```

5 Generate some activity on the demo Adabas database. For example, run *getdbinfo <dbid>*, or run the *c_example.exe* on your database (*c_example.exe* is located in the */bin/examples* subdirectory of *AdabasClient*)

This step sends event data to Elasticsearch, and allows Elasticsearch to build the initial index structure.

6 Access Kibana from a browser (for example Firefox or Chrome) with the following URL:

```
http://<KIBANA_HOSTNAME>:5601
```

7 Create an index pattern for your event data:

1. In Kibana, go to **Management**->**Index Patterns**->**Add new**.
2. Keep the checkbox **Index contains time-based events** checked.
3. Enter the index name *adabas_analytics-**.
4. Wait for a few seconds and leave the time field selection at **event_timestamp**.
5. Click on the **Create** button.

Note: You can validate this step by going to **DevTools**->**Console** and entering the command `GET _cat/indices`. This should produce output of the following form:

```
...
... adabas_analytics-<CURRENT_DATE> ...  <CURRENT_SIZE>
... .kibana                         ...  <CURRENT_SIZE>
... ↩
```

8 You can now display your data in Kibana by going to **Discover**, and then selecting the index**adabas_analytics-*** from the drop-down box.

Note: If you don't see event data, try setting a longer period under review. In the top right corner of your Kibana browser window, click on **last 15 minutes** and select the timeframe for which you want to review the data.

9 Import the predefined Adabas/Natural dashboard for Kibana, as well as the visualization objects located in the directory *AdabasAnalytics/third-party/KibanaExampleDashboard*. In Kibana, go to **Management**->**Saved Objects**->**Import** and select the delivered json files (one at a time).

Note: The visualization objects can only be imported if index data already exists in ElasticSearch. A corresponding message is displayed if problems occur.

Troubleshooting

If an error occurs, check the contents of the *ealserver.log* file in the *log* subdirectory of your Adabas-Analytics installation. Look for the entry "Elastic Search Sink "ElasticSearch" sending events to cluster <YOUR_CLUSTER_NAME>" where *YOUR_CLUSTER_NAME* matches the *cluster.name* entry in the *elasticsearch.yml* configuration file. If the log file contains the above entry and if the problem still persists, check your firewall settings.

If Elasticsearch doesn't start on your system, check the log files in the *logs* directory under your <*ELASTICSEARCH_INSTALL_DIR*> for configuration errors. If any configuration error is fixed

but Elasticsearch still refuses to start, the Elasticsearch documentation recommends that you deactivate the system call filters at your own risk. This is done by setting *bootstrap.system_call_filter* to false in the *elasticsearch.yml* config file.

Frequently Asked Questions

How can I share dashboards without Kibana administration features?

1. Click on **Dashboard** in the side navigation.
2. Open the dashboard you want to share.
3. Add *&embed=true* at the end of the address line of your Kibana dashboard URL.
4. Click on **Share**.
5. Copy the link you want to share. We recommend that you use use shortened snapshot URL.

How can I protect the Kibana index located in ElasticSearch?

1. Click on **Dev Tools** in the side navigation.
2. Execute the console command *GET _cat/indices* to get a list of existing indices.
3. Execute the console command *PUT /<yourKibanaIndexName>/_settings {"index.blocks.read_only": true}* to disable any modifications to your Kibana index.
4. Execute the console command *GET /<yourKibanaIndexName>/_settings* to check your current settings.

Note: In case of protection, a fatal error message will be returned

```
"Request to Elasticsearch failed: ...
                ... [FORBIDDEN/5/index read-only (api)];"
```

How can I backup and restore an individual ElasticSearch index?

1. Refer to the section *Elasticsearch Reference [5.1]* | *Modules* | *Snapshot And Restore* in the online documentaton **https://www.elastic.co/guide/en/elasticsearch/reference/5.1/modules-snapshots.html**.

How can I delete a daily index generated by receiving events from the Adabas Analytics Server?

1. Click on **Dev Tools** in the side navigation.
2. Execute the console command `GET _cat/indices/adabas_analytics-*` to get a list of existing daily Adabas Analytics indices.
3. Execute the console command `DELETE /adabas_analytics-<yourSelectedDailyIndex>`.
4. Execute the console command `GET _cat/indices/adabas_analytics-*` again to validate the result.

Note: Refer to the section *Elasticsearch Reference [5.1]* | *Indices APIs* | *Delete Index* in the online documentation.

How can I develop my own dashboards?

We recommend that you refer to Kibana's "Getting Started" Tutorial - see **https://www.elastic.co/guide/en/kibana/current/getting-started.html**

8 Adabas Extensions for Adabas Analytics

The current version of Adabas includes the following extensions, which enable you to work more easily with Adabas Analytics:

Extention	Purpose
ADAELP	Event log report. Used to print events from the Adabas Anlytics event log.
EALCONFIG	Event Analytics configuration tool. Used to help you set up the Adabas Analytics component.

ADAELP (Event Log Report)

This section describes the utility "ADAELP".

Functional Overview

The ADAELP utility prints events from an event log created by Adabas Analytics.

Note: Event logging must be enabled and the replication user exit must be loaded in order to write event logs. For further information see the section *Concepts*.

The ADAELP parameters USER_ID, HOSTNAME and EVENT_TIMESTAMP select a subset of the events in the event log.

In the interactive mode, ADAELP displays the selected events when the keyword LIST is entered. If ADAELP is called with parameters, the selected events are displayed immediately.

Events are displayed as follows: a first line with the event type is followed by lines that contain the field data of the event in question. The display of an event is concluded with the event type being repeated on the last line.

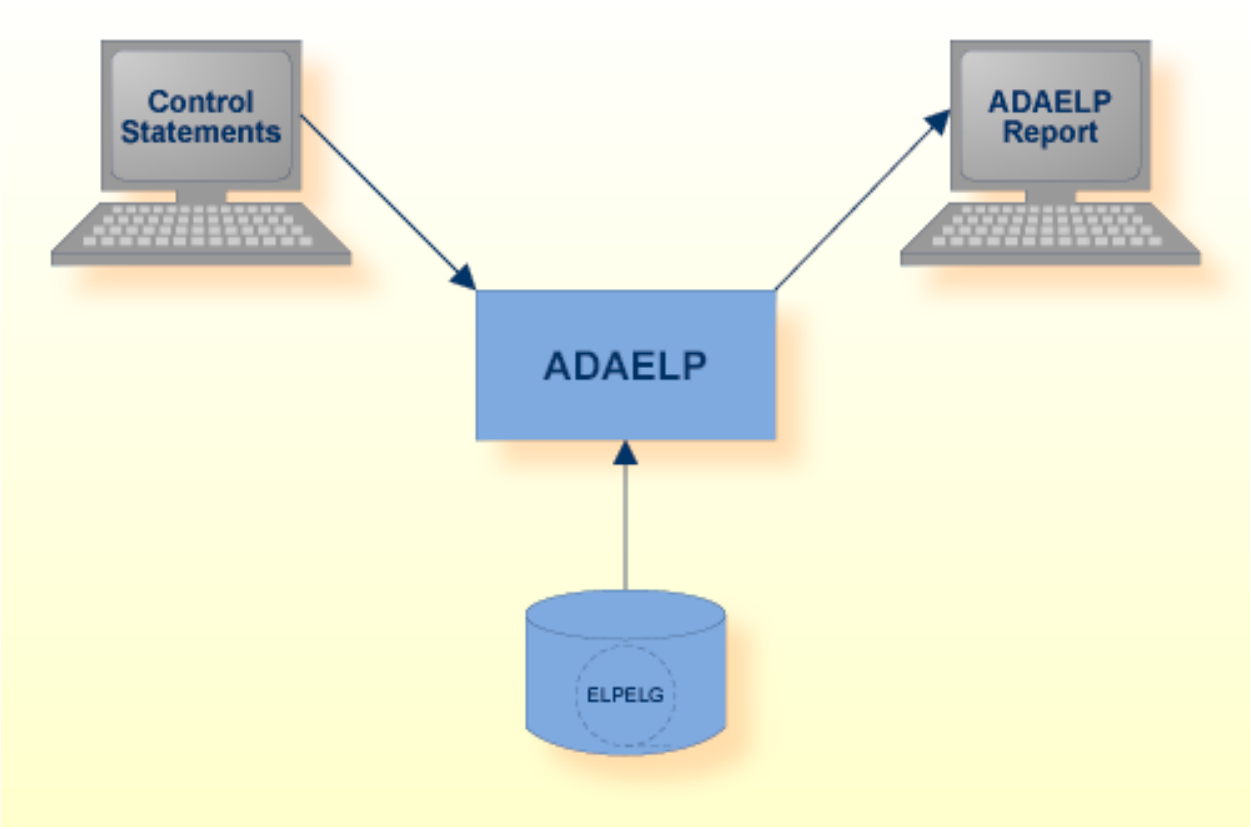
Example output

```
start read event
  event_timestamp=16-JUL-2015 11:51:01.977020
  dbid=163
  file_number=1
  command_code=L5
  response_code=0
  isn=993
  pid=6772
```

```
  hostname=PCST01
  user_id=st
  tsid=68 ba 56 02 fb 1a 05 00
end read event
```

This utility is a single-function utility.

Procedure Flow



Data Set	Environment Variable/ Logical Name	Storage Medium	Additional Information
Event log	ELPELG	Disk	
Control statements	stdin		see section Control Parameters
ADAELP report	stdout		

Checkpoints

The utility writes no checkpoints.

Control Parameters

The following control parameters are available:

```
D    DBID = number

    EVENT_TIMESTAMP = ([absolute-date][,[absolute-date]])

    HOSTNAME = string

    LIST

    USER_ID = string
```

DBID

```
DBID = number
```

This parameter specifies the database ID of the database for which the event log was written.

EVENT_TIMESTAMP

```
EVENT_TIMESTAMP = ([absolute-date][,[absolute-date]])
```

This parameter selects the log records in the range specified by the optional date strings. The date strings must correspond to the following absolute date and time format:

```
dd-mmm-yyyy[:hh:mm:ss[.mmmmmm]]
```

Leading zeroes in the date and time specification may be omitted. Any numbers not specified are set to 0, for example 28-jul-2015 is equivalent to 28-jul-2015:00:00:00.000000.

By default, all log records are selected.

Examples:

```
adaelp: event_timestamp = 8-aug-2015
```

The event with event_timestamp 8-AUG-2015 00:00:00 is selected.

```
adaelp: event_timestamp = (8-aug-2015:12,)
```

All events with time_stamp from 8-AUG-2015 12:00:00 onwards are selected.

```
adaelp: event_timestamp = (,8-aug-2012:12:34)
```

All events with time_stamp before 8-AUG-2015 12:34:00 are selected.

```
adaelp: event_timestamp = (16-JUL-2015 11:51:01.977020, 16-JUL-2015 11:51:02.177000)
```

All events with event_timestamp from 16-JUL-2015 11:51:01.977020 to 16-JUL-2015 11:51:02.177000 are selected.

HOSTNAME

```
HOSTNAME = string
```

This parameter selects all events with the hostname specified by 'string'. The length of the parameter value is limited to 8 characters.

LIST

```
LIST
```

This parameter lists the events selected with the parameters DBID, EVENT_TIMESTAMP, HOSTNAME and USER_ID.

USER_ID

```
USER_ID = string
```

This parameter selects all events with the user ID specified by 'string'. The length of the parameter value is limited to 8 characters.

Specifying Multiple Selection Criteria

If multiple selection criteria are specified, they are combined by a logical AND, e.g.

```
event_timestamp=(8-aug-2015:12:34,), user_id = guest, hostname = machine3
```

This selects all events after 8-aug-2015:12:34 with user_id = guest and hostname = machine3.

EALCONFIG (Event Analytics Configuration Tool)

A simple configuration tool (EALCONFIG) is provided to help you set up the Adabas Analytics component. All configuration parameters are stored in the appropriate *DBnnn.INI* file in the database directory. The database with a valid *DBnnn.INI* file must exist.

Compared to the version provided with the Adabas Version 6.5.0 package, the new version of EALCONFIG has the following major differences:

- New subtopics TARGET_EAL_SERVER and TARGET_NUCELG;
- The items SWITCH_AFTER_EVENTS and SWITCH_AFTER_TIME are moved to TARGET_NUCELG;
- The item TARGET_NUCELG is ignored if TARGET_EAL_SERVER exists;
- The new event types NAT_INSERT, NAT_READ, NAT_UPDATE, NAT_DELETE, NAT_COMMIT, NAT_ROLLBACK, ADA_PERF and ADA_NAT_PERF are supported.

Notes:

1. To delete the TARGET_EAL_SERVER entry from *DBnnn.INI*, execute the command `adaini dbid=<nnn> delete topic=event_analytics topic=TARGET_EAL_SERVER item=*`.
2. To display the EVENT_ANALYTICS entries, execute the command `adaini dbid=<nnn> show topic=event_analytics`.

Using EALCONFIG

The configuration tool will ask you for each parameter one at a time. It does not accept command line parameters.

The following parameters are requested by the configuration tool. The current settings or default values are shown in brackets ([]). Pressing the Enter key without any input will keep the current setting.

Dbid
The database ID of the database to be configured for Adabas Analytics.

Activate
Enable or disable the event logging. Possible values are *yes* and *no*, the default is *yes*.

Destination of Events

Choose *SERVER* if you want to send the events to the Adabas Analytics server. Choose *FILE* if you want to send the events to a file. The default is *SERVER*.

File name (only if destination *FILE* is selected)

The fully-qualified path name for the log file. The default is *$ADADATADIR/dbnnn/NUCELG* or *%ADADATADIR%\dbnnn\NUCELG*.

Filter events for files

A list of database file numbers for which the event logging is to be performed. For multiple values, specify the list surrounded by brackets, e.g. '(10,11,12,30-40,100)'; for all files enter an asterisk '*'. The default value is '*'.

Filter for events

A list of event types which will be logged. The default is all events. For multiple values, specify the list surrounded by brackets, e.g. '(INSERT,UPDATE,COMMIT)', for all events enter an asterisk '*'. For further information about the supported event types, see ***Adabas Analytics Event Types***.

Switch log file after events (only if destination *FILE* is selected)

The number of events that occur before starting a new log file. The default is never to switch.

Switch log file after time (only if destination *FILE* is selected)

The number of seconds that elapse before starting a new log file. The default is never to switch.

Host name (only if destination *SERVER* is selected)

This is the host name where the analytics server is running. The default is *localhost*.

Port number (only if destination *SERVER* is selected)

This is the port number that the analytics server listens to. The default is *6521*.

Reconnect timeout (only if destination *SERVER* is selected)

If the connection between the Adabas nucleus and the analytics server is lost, this parameter specifies the time in seconds that the Adabas nucleus waits until it tries to reconnect to the analytics server. The default is *1*.

Maximum number of reconnect attempts (only if destination *SERVER* is selected)

If the connection between the Adabas nucleus and the analytics server is lost, this parameter specifies the number of reconnect attempts. The default is *0*, which means 'try forever'.

Note: If the maximum number of attempts is not 0, and if the maximum number of reconnect attempts is reached without reconnecting successfully, event logging will be deactivated.

Nucleus behaviour in case of error (only if destination *SERVER* is selected)

Select *ABORT* if you want to abort the Adabas nucleus if a connection error occurs between the Adabas nucleus and the analytics server. Select *IGNORE* if you want to continue with the Adabas nucleus if a connection error occurs between the Adabas nucleus and the analytics server.

Example 1 (destination for the events is *FILE*)

The following example shows a first run of the configuration tool. The result of this configuration will be: for the files 10,11,12,30 to 40 and 100, the events for insert, update and commit commands are logged in the file */opt/softwareag/Adabas/db199/NUCELG.0001*. After 86400 seconds (i. e. 24h) or after 1000000 events (depending on which occurs first), the current *NUCELG* file will be closed and a new file with an increased suffix will be created (*NUCELG.0002*, *NUCELG.0003*, ...).

```
$> ealconfig

Adabas Analytics Configuration Tool

Current/default values are shown in '[]', press <enter> to keep these values.
Multiple values have to be enclosed in brackets.

Please enter the dbid: 199
Do you want to activate event logging [YES|NO]? [YES] yes

Please enter the file name for the EAL log [/opt/softwareag/Adabas/db199/NUCELG]:

Filter events for files. Only for the specified files the events are logged.
Format: (5,8,10,20-30,40-50), (*) for all files.
Please enter the file list [*]: (10,11,12,30-40,100)

Filter events. Only the specified events are logged.
Available event types:
   (INSERT, READ, UPDATE, DELETE, COMMIT, ROLLBACK,
    NAT_INSERT, NAT_READ, NAT_UPDATE, NAT_DELETE, NAT_COMMIT, NAT_ROLLBACK,
    ADA_PERF, ADA_NAT_PERF),  (*) for all events.
Please enter the event type list [*]: (INSERT,UPDATE,COMMIT)

Log file switching. You can begin a new log file after a number of events occurred ↩
and/or after a certain time.
Please enter the number of events ('0' - do not switch) []: 1000000
Please enter the number of seconds ('0' - do not switch) []: 86400

Saving changes to DB199.INI file.

Terminated.
```

The resulting *DB199.INI* file then contains the following entry:

```
[EVENT_ANALYTICS]
ACTION                   = YES
[TARGET_NUCELG]
    SWITCH_AFTER_EVENTS      = 1000000
    SWITCH_AFTER_TIME        = 86400
[TARGET_NUCELG-END]

  [FILTER]
```

```
    FILES                   = (10,11,12,30-40,100)
    EVENT_TYPES             = (INSERT,UPDATE,COMMIT)
  [FILTER-END]
[EVENT_ANALYTICS-END]
```

Example 2 (destination for events is *SERVER*)

The following example shows a first run of the configuration tool. The result of this configuration will be: for the files (9,11,202-205), the ADA_NAT_PERF event is logged and sent to the analytics server. The analytics server is running on localhost, the port number is 6521. If the connection between the Adabas nucleus and the analytics server is lost, the Adabas nucleus ignores the disconnect, but still tries to reconnect to the analytics server every second.

```
$> ealconfig

Adabas Analytics Configuration Tool (v6.5.1)

Current/default values are shown in '[]', press <enter> to keep these values.
Multiple values have to be enclosed in brackets.

Please enter the dbid: 199
Do you want to activate event logging [YES|NO]? [YES]

Destination of Events: You can send events to the Adabas Analytics Server,
the default, or write the events into a file.
Which destination do you want [SERVER|FILE]? [SERVER]

Filter events for files. Only for the specified files the events are logged.
Format: (5,8,10,20-30,40-50), (*) for all files.
Please enter the file list [(9,11,202-205)]:

Filter events. Only the specified events are logged.
Available event types are:
   (INSERT, READ, UPDATE, DELETE, COMMIT, ROLLBACK, NAT_INSERT, NAT_READ,
    NAT_UPDATE, NAT_DELETE, NAT_COMMIT, NAT_ROLLBACK, ADA_PERF, ADA_NAT_PERF),
    (*) for all events.
Please enter the event type list [*]: ADA_NAT_PERF
Please enter the host name where the analytics server is running [localhost]:
Please enter the port number of the analytics server [6521]:
Please enter the reconnect timeout (seconds) [1]:
Please enter the maximum number of reconnect attempts ('0' - try forever) [0]:
The behaviour of the nucleus for errors with the analytics server.
The nucleus can ABORT or just ignore the error and try to reconnect to the
Analytics Server.
Please enter one of the available actions: ABORT, IGNORE [IGNORE]:

Saving changes to DB199.INI file.

Terminated.
```

The resulting *DB199.INI* file then contains the following entry:

```
[EVENT_ANALYTICS]
  ACTION               = YES
  [FILTER]
    EVENT_TYPES          = ADA_NAT_PERF
    FILES                = (9,11,202-205)
  [FILTER-END]
  [TARGET_EAL_SERVER]
    HOST                 = localhost
    MAX_RETRIES          = 0
    ON_ERROR             = IGNORE
    PORT                 = 6521
    RECONNECT_TIMEOUT    = 1
  [TARGET_EAL_SERVER-END]
 [EVENT_ANALYTICS-END]
```

Parameters of Adabas Analytics in the Configuration File DBnnn.INI

Adabas Analytics uses a new topic ([EVENT_ANALYTICS]) in the configuration file *DBnnn.INI*. An example of how this topic might look is shown in the examples above.

This section describes the keywords and subtopics contained in the [EVENT_ANALYTICS] topic.

Note: All of the keywords and subtopics are optional.

ACTION = YES/NO

This keyword activates/deactivates Adabas Analytics. If this keyword is omitted, Adabas Analytics is switched off. Valid keywords are YES and NO.

Subtopic TARGET_NUCELG

SWITCH_AFTER_EVENTS = <count>

This keyword is used to switch to a new *NUCELG* file after <count> events have been written to the *NUCELG* file. If this keyword is omitted, all events are logged to a single *NUCELG* file.

SWITCH_AFTER_TIME = <time in seconds>

This keyword is used to switch to a new *NUCELG* file after <time in seconds> has passed and a new event is to be generated. If this keyword is omitted, all events are logged to a single *NUCELG* file.

Subtopic FILTER

The subtopic FILTER has 2 keywords: FILES and EVENT_TYPES. These keywords are used to determine which types of events are generated for which Adabas files.

FILES = (<file1>, <file2>, <file3> - <file4>)

This keyword specifies the Adabas files that are to be active for eventing. You can specify a list of files; the list can contain single files (<file1>, <file2>) or a range of files (<file3> - <file4>). If this keyword is omitted, events are generated for all Adabas files in the database.

EVENT_TYPES = (INSERT,READ,UPDATE,DELETE,COMMIT,ROLLBACK, NAT_INSERT, NAT_READ, NAT_UPDATE, NAT_DELETE, NAT_COMMIT, NAT_ROLLBACK, ADA_PERF, ADA_NAT_PERF)

This keyword specifies the types of events that are generated. The details of each event type are described in the section ***Adabas Analytics Event Types***. If this keyword is omitted, all event types are generated for the files specified by the FILES keyword.

Adabas Analytics Filter Mechanism

The filter mechanism gives you control over the events that are generated.

You can filter the events by the following criteria:

- Adabas file number;
- Event type.

You can filter by Adabas file number only, by event type only or you can combine both filters. For further information about the syntax and semantics of the filters for FILES and EVENT_TYPES, see the section ***Parameters of Adabas Analytics in the Configuration File DBnnn.INI***.

The file *DBnnn.INI* contains the topic EVENT_ANALYTICS and the subtopic FILTER.

Example entry in DBnnn.INI:

```
[EVENT_ANALYTICS]
  ACTION                  = YES
[TARGET_NUCELG]
    SWITCH_AFTER_EVENTS     = 1000000
    SWITCH_AFTER_TIME       = 86400
[TARGET_NUCELG-END]

  [FILTER]
    FILES                 = (10,11,12,30-40,100)
    EVENT_TYPES           = (INSERT,UPDATE,COMMIT)
  [FILTER-END]
[EVENT_ANALYTICS-END]
```

This entry results in events being generated for the Adabas files 10,11,12,30-40 and 100. The events are limited to the types insert, update and commit. After 86400 seconds (24 hours) or after 1000000 events (depending on which comes first), the current *NUCELG* file will be closed, and a new file with an increased suffix will be created (NUCELG.0002, NUCELG.0003 ...).

Adabas Analytics Event Types

Adabas Analytics currently supports 7 types of conceptual distinct events in two categories. Six events belong to the category auditing, and one event belongs to the category performance monitoring. For each of the 7 conceptual events, there are two manifestations: events with/without Natural-specific information, thus resulting in an overall of 14 distinct event types. Events containing Natural information are prefixed with “NAT_”, events without Natural information do not have a prefix.

INSERT
A new record has been created, triggered by Adabas commands of the type Nx.

NAT_INSERT
Same as INSERT, but with additional Natural information.

READ
A record has been searched, triggered by Adabas commands of the types Lx and Sx.

In the case of Read commands (Lx) with the multifetch option specified, a read event will be generated for each record that can be read. If a record cannot be read because it is held exclusively by another user, a read event with response code *145* will be triggered for this record, but only if the return option (*0*) was specified.

NAT_READ
Same as READ, but with additional Natural information.

UPDATE
A record has been modified, triggered by an Adabas command of the type A1.

NAT_UPDATE
Same as UPDATE, but with additional Natural information.

DELETE
A record has been removed, triggered by an Adabas command of the type E1.

NAT_DELETE
Same as DELETE, but with additional Natural information.

COMMIT
A transaction has been committed, triggered by Adabas commands of the types ET and CL.

NAT_COMMIT
Same as COMMIT, but with additional Natural information.

ROLLBACK
A transaction has been backed out, triggered by an Adabas command of the type BT, and also by any other command that results in the nucleus response code 9.

NAT_ROLLBACK
Same as ROLLBACK, but with additional Natural information.

ADA_PERF
A command was processed by Adabas. This event contains additional information regarding the command, such as time taken and command options.

NAT_ADA_PERF
Same as ADA_PERF, but with additional Natural information.

The fields in the event types (which can be displayed using the utility ADAELP) contain the following information:

Field	Information Contained
event_timestamp	Creation time of this event.
dbid	Adabas database ID.
file_number	Adabas file number.
command_code	Adabas command code.
response_code	Adabas response code.
isn	Adabas ISN.
pid	Process ID of the Adabas client.
hostname	Machine name of the Adabas client.
user_id	User ID of the Adabas client.
tsid	Unique marker of the Adabas client.
natapplication*	Name of the Natural application issuing the call.
natprogram*	Name of the Natural program issuing the call.
natcount*	Number of ADABAS calls since the last IO.
natexec*	Number of times a Natural object has been executed (internal).

Field	Information Contained
natlevel*	Natural call level of the program executed.
natuser*	The Natural user issuing the call.
natstatement*	Natural statement number.
natlib*	The Natural library issuing the call.
natrpcclientuid*	Natural RPC client user ID.
natrpcid*	Natural RPC ID.
natrpcconvid*	Natural RPC conversation ID.
natsecgroup*	Natural security group.
additions1	Contains the additions1field of the issued call.
command_duration	Contains the time taken, in microseconds, that Adabas took to process the call
copt1	Contains the value of command option 1 of the processed call
copt2	Contains the value of command option 2 of the processed call

*) These fields are only available when using events with Natural information.