jACOB™ Administration

Manual

jACOB™ Administration: Manual

Published 2005

Table of Contents

1. The Engine domain 1
    1. Deploying applications, versions and patches 2
        1. Deploying a new application version 5
        2. Deploying a patch 7
    2. Defining data sources 8
        1. Defining a JNDI data source 11
        2. Defining a jACOB™ maintained data source without connection pooling 12
        3. Defining a jACOB™ maintained data source with connection pooling 14
        4. Reconfiguring data sources 16
        5. Setting up data sources 18
    3. Configuring the jACOB™ application server 20
        1. Monitoring properties 24
        2. jAN properties 26
        3. Browser properties 27
        4. Internationalization properties 28
        5. Toolbar properties 29
        6. Miscellaneous properties 30
    4. Administrating user sessions and jACOB™ administrator accounts 33
        1. Retrieving currently active user sessions 35
        2. Terminating user sessions 36
        3. Creating new jACOB™ administrator accounts 37
        4. Changing the password of a jACOB™ administrator 38
        5. Resetting the password of a jACOB™ administrator 39
    5. Administrating locks on data records 40
        1. Removing a record lock 41
    6. Administrating jACOB tasks 42
        1. Viewing jACOB tasks with an incorrect last execution 44
    7. Administrating reports 45
        1. Viewing all reports scheduled by a particular user 47
    8. Licensing the jACOB™ application server 48
        1. Installing a new license key 49
2. The Messaging domain 50
    1. Administrating messages 51
        1. Viewing messages with an error message status 53
3. The Monitoring domain 54
    1. Monitoring SQL data sources 55
        1. Viewing the time consuming SQL statements of an application 56
    2. Monitoring system resources 57
        1. Viewing the memory requirements 59

Glossary of terms 60

List of Figures

1.1. Engine domain 1
1.2. Applications form 3
1.3. JNDI data sources and jACOB™ maintained data sources 8
1.4. Data Sources form 9
1.5. Setup of data source 18
1.6. Properties form 21
1.7. Toolbar buttons 29
1.8. Users form 33
1.9. Locks form 40
1.10. Tasks form 42
1.11. Reports form 45
2.1. Messaging domain 50
2.2. jAN form 51
3.1. Monitoring domain 54
3.2. SQL form 55
3.3. System form 57
3.4. Excel chart of the memory requirements in the past 59

List of Tables

1.1. Predefined properties 23
1.2. Toolbar properties 29

List of Examples

1.1. Priority hierarchy of properties 20
1.2. Property application.default 30
1.3. Property gui.window.prefix 31
1.4. Property user.theme.default 32

Chapter 1. The Engine domain

Introduction

The Engine domain is the starting point for the administration and configuration of jACOB™ applications and of the application server itself.

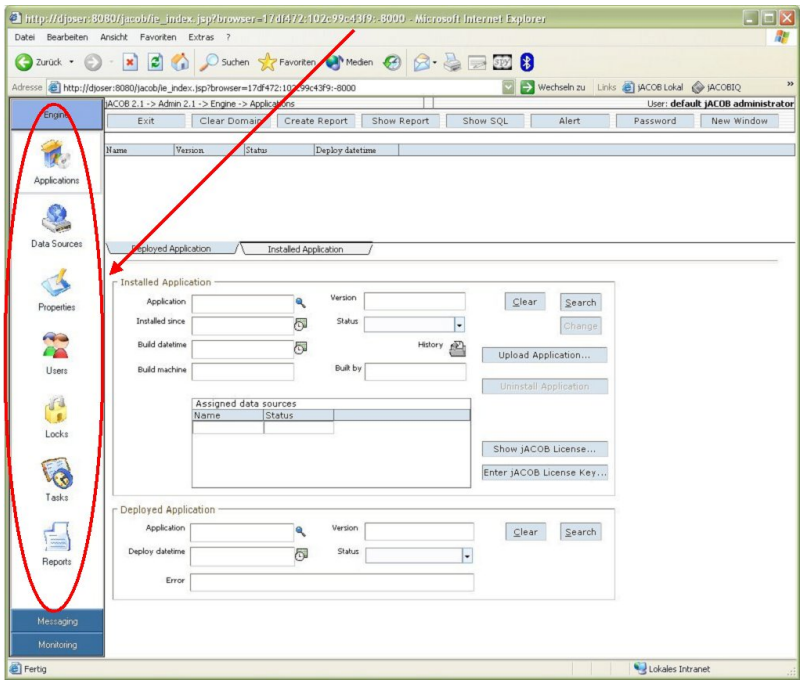


Figure 1.1. Engine domain

What takes place in the Engine domain?

The following actions take place in the Engine domain:

- Deploying applications, versions and patches
- Defining data sources
- Configuring the jACOB™ application server
- Administrating user sessions and jACOB™ administrator accounts
- Administrating locks on data records
- Administrating jACOB tasks
- Administrating reports
- Licensing the jACOB™ application server

Deploying applications, versions and patches

Introduction

The jACOB™ Administration enables *hot deployment* of :

- New applications (first version)
- New versions of applications already deployed and
- Patches to correct software errors in applications already deployed.

When an application version or patch is deployed four actions take place sequentially:

1. The jACOB™ Administration uploads the packed "*.jacapp" file of the application version onto the jACOB™ application server.
2. The jACOB™ application server unpacks the "*.jacapp" file.
3. The content of the packed "*.jacapp" file is written to the jACOB configuration data source.
4. The application version or patch is registered as being deployed on the jACOB™ application server.

After this the new application version or patch is available on the jACOB™ application server.

Running jACOB™ in a cluster

jACOB™ can be also run in a cluster.

Action 3 "installs" the application version in the jACOB™ configuration data source. This makes the application version available for further jACOB™ application server nodes within the same cluster. To guarantee conformance, each node of the cluster periodically synchronizes itself with the configuration data source and deploys new application versions, automatically.

This implies that at any one moment in time an installed application version can be deployed on one server node while it is not deployed yet on another server node.

Form and domain

The deployment of application versions and patches takes place within the Applications form of the Engine domain.

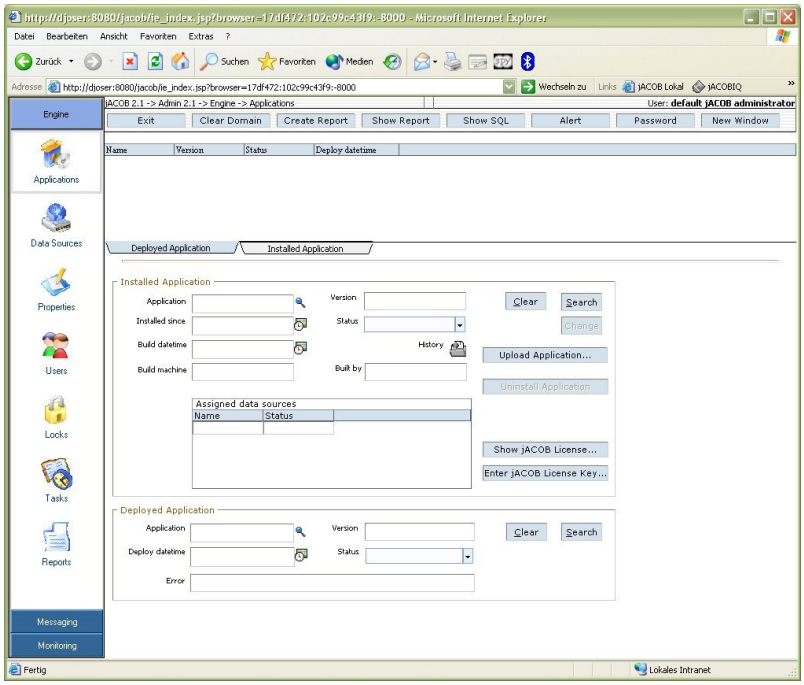


Figure 1.2. Applications form

The form information is devided into the following groups:

- Installed Application
- Deployed Application

Installed applications

The group Installed Application shows all jACOB™ application versions "installed" in the jACOB™ configuration data source. An installed application version has the following data structure:

- Name
- Version
- Installed since (the date the last application version or patch was installed)
- Status of the application version
- Build information such as the build date, the machine it has been built on and the ID of the application programmer who initiated the build.
- A list of all Assigned data sources and relevant Status, assigned to the application version.

Deployed applications

The group Deployed Application shows all jACOB™ application versions deployed on the jACOB™ application server. A deployed application version has the following data structure:

- Name

- Version
- Deploy datetime
- Deploy Status of the application version (Success or Error)
- Error message in case of Status Error, i.e. the deployment has not been successful

Status of an installed application version

The Status of an installed application version is the only field in the form that can be modified. The Status defines whether:

- The application version is inactive for all users,
- The application version is productive or
- The application version is a test version visible to test users only.

The initial status of an application version is inactive. This means that the application version is deployed, but no user sessions or tasks can be run.

Status of the Assigned data sources

The Status of the Assigned data sources can be:

- undefined, i.e. the data source has not been completely specified yet,
- defined, i.e. the data source is already completely specified, or
- predefined, i.e. the jACOB™ application server recognizes the data source as a jACOB™ internal database.

Note

The settings of predefined data sources cannot be modified through the jACOB™ Administration. Therefore, predefined data sources do not appear in the Data sources form of the Engine domain.

Running different versions in parallel

The jACOB™ application server supports the parallel running of different versions of one application.

This for instance enables the deployment of a new version on a productive platform to be visible to a restricted number of persons only.

Uninstalling applications

jACOB™ application versions can be uninstalled by pressing the Uninstall Application button in the Installed Application group of the Applications form.

By doing so, the jACOB™ application server cancels the application version. The application version is removed from the jACOB™ configuration data source. If jACOB™ is setup as a cluster, other server nodes within the same cluster will remove the application version at the next periodical synchronization with the configuration data source.

Procedures

The following procedures will be described in the next sections:

- Deploying a new application version
- Deploying a patch

Deploying a new application version

Purpose

Before running a new application or new version of an already deployed jACOB™ application the application version must first of all be deployed onto the jACOB™ application server.

This can be done with help of the *hot deployment* functionality in the jACOB™ Administration.

Impact

This functionality includes:

- The upload of the packed "*.jacapp" file containing the application version to be deployed onto the server,
- The unpacking of the "*.jacapp" file and
- The registration of the application version on the jACOB™ application server.

Hereafter, the application version is available on the jACOB™ application server. The data of the application version is shown within the data fields of the Application form.

Initially the application version has the Status inactive. Setting the application version Status to productive will make it available to all users.

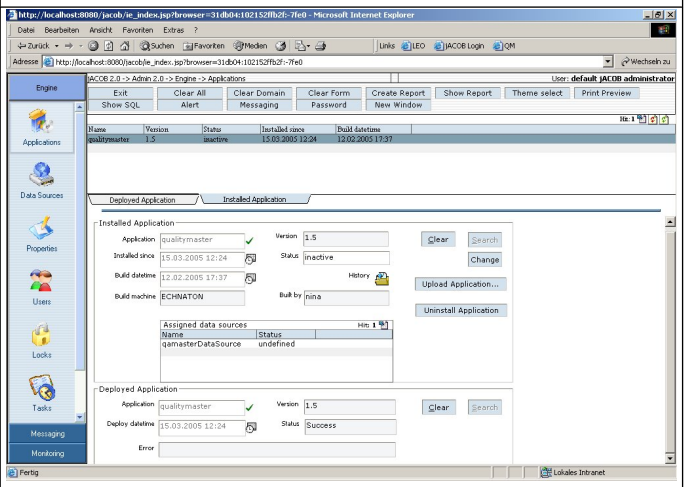
The table Assigned data sources lists all data sources and relevant Status assigned to the application version. If a new application has been deployed, most Assigned data sources will be new to the jACOB™ application server and therefore, the initial Status will be undefined.

The next step is to define the undefined data sources.

Procedure

Perform the following steps to deploy a new application version onto the jACOB™ application server:

Step	Procedure
1	Open the Applications form of the Engine domain.
2	Press the button Upload Application.... After this a dialog box will pop up asking to specify the "*.jacapp" file containing the application version to be deployed.
3	Specify the "*.jacapp" file and press the button Save. Hereafter, the file is uploaded and unpacked. The application version is registered and made available on the jACOB™ application server. The data fields of the Application form now contain the data of the deployed application version.

Step	Procedure
	
4	To define the undefined Assigned data sources click on the undefined data sources in the table. When following the above procedure, you will get to the Data Source form of the Engine domain. Continue with the Defining data sources procedure.

Deploying a patch

Purpose

Software errors can be corrected by deploying a patch.

Similar to the deployment of applications versions this can be done by using the *hot deployment* functionality in the jACOB™ Administration.

Impact

As already described in detail in Deploying a new application version, this functionality also includes the upload and unpacking of the "*.jacapp" file containing the patch.

However, contrary to the deployment of application versions, a patch will be activated immediately after the next user request to the application server.

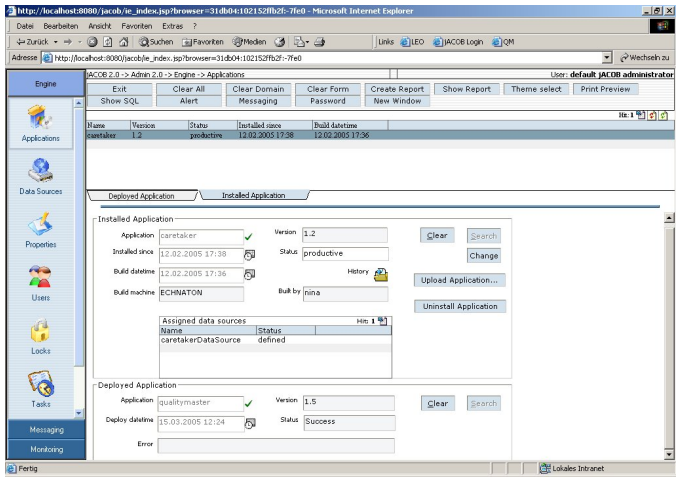
The patch inherits the Status of the error version.

Note

If jACOB™ is setup as a cluster, the other application server nodes will deploy the patch during the next periodical synchronization with the jACOB™ configuration data source.

Procedure

Perform the following steps to deploy a patch onto the jACOB™ application server:

Step	Procedure
1	Open the Applications form of the Engine domain.
2	Press the button Upload Application.... After this a dialog box will pop up asking you to specify the "*.jacapp" file containing the patch to be deployed.
3	Specify the "*.jacapp" file and press the button Save. Hereafter, the file is uploaded and unpacked. At the next user request to the application server, the patch will be activated. The data fields of the Application form are now filled out with the data of the patch deployed. 

Defining data sources

Introduction

After an application has been deployed most of its assigned data sources will be undefined, i.e. unknown to the jACOB™ application server. Undefined data sources have to be defined before running the application. This can be done with the define data source functionality in the jACOB™ Administration.

The following types of data sources can be defined:

- JNDI data sources,
- jACOB™ maintained data sources without *connection pooling* and
- jACOB™ maintained data sources with *connection pooling*

JNDI and jACOB™ maintained data sources

JNDI data sources are data sources that are located via *JNDI*. JNDI data sources may be shared between jACOB™ and third party applications within the same web server instance.

Whereas jACOB™ maintained data sources are data sources that are maintained by the jACOB™ application server. jACOB™ maintained data sources may only be shared between jACOB™ applications.

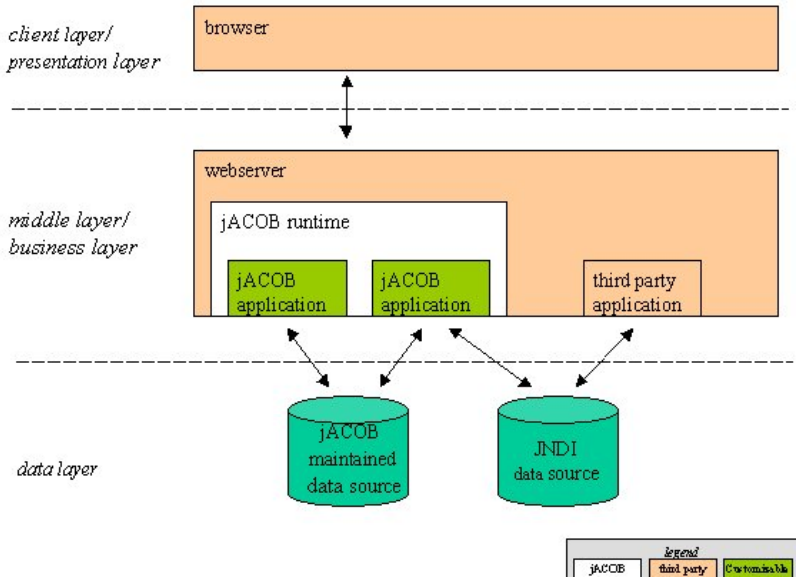


Figure 1.3. JNDI data sources and jACOB™ maintained data sources

Form and domain

Data sources are defined within the Data Sources form of the Engine domain.

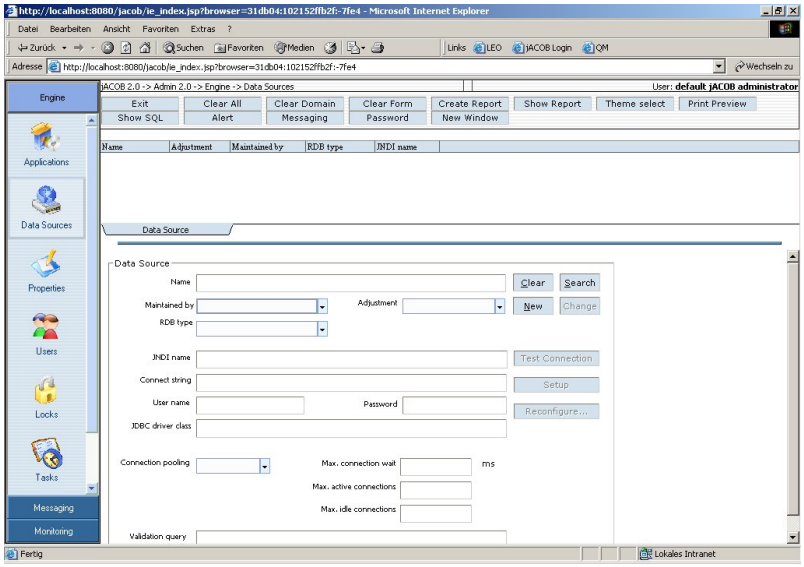


Figure 1.4. Data Sources form

The Data Source group shows the following information about a data source:

- Name
- The application server the data source is Maintained by
- RDB type, specifying the type of data source
- Adjustment to jACOB or Quintus conformity
- JNDI name in case of a *JNDI data source*
- Connect string to be passed to the JDBC driver
- User name and Password of the owner/user (with full access rights) to be passed to the JDBC driver
- JDBC driver class name of the JDBC driver to be used
- Is Connection pooling enabled?
- Max. connection wait time that the *connection pool* will wait (when there are no available connections) for a connection to be returned
- Max. active connections that can be allocated from the *connection pool* and Max. idle connections in the *connection pool* at the same time
- Validation query used by the *connection pool* to validate connections before these are returned to the application

The Maintained by field

jACOB™ maintained data sources are Maintained by the jACOB application server.

Whereas *JNDI data sources* are Maintained by the Webserver itself.

If the data source is a *JNDI data source*, its JNDI name will have to be entered in the JNDI name field.

In case of *jACOB™ maintained data sources* the Connect string, User name, Password and JDBC driver class must be specified.

RDB type of a data source

The RDB type specifies the type of data source. jACOB™ currently supports the following data source types:

- Oracle,
- MSSQL,
- HSQL and
- AutoDetect if the data source is a *JNDI data source*.

The values of the Connect string and JDBC driver class are subject to special patterns. These depend on the specified RDB type. The jACOB™ Administration enters the patterns into these fields by default.

Adjustment

The Adjustment field defines how e.g. data records are locked, new keys are generated and record modifications are historicized. This for instance is required when running a jACOB™ and a Quintus application in parallel without migrating the data sources.

Adjustment to the following systems are currently supported:

- jACOB and
- Quintus

If a data format does not have to be compatible to a legacy third party application, Adjustment should be set to the default value jACOB.

Reconfiguration of data sources

The deployment of new application versions sometimes requires the reconfiguration of the data sources assigned to the application. This is the case if the structure of the data sources is not in line with the structure expected by the application.

Procedures

The following procedures will be described in the next sections:

- Defining a JNDI data source
- Defining a jACOB™ maintained data source without connection pooling
- Defining a jACOB™ maintained data source with connection pooling
- Reconfiguring data sources
- Setting up data sources

Defining a JNDI data source

Purpose

This section describes how undefined data sources that are located via *JNDI* are defined within the jACOB™ Administration.

Impact

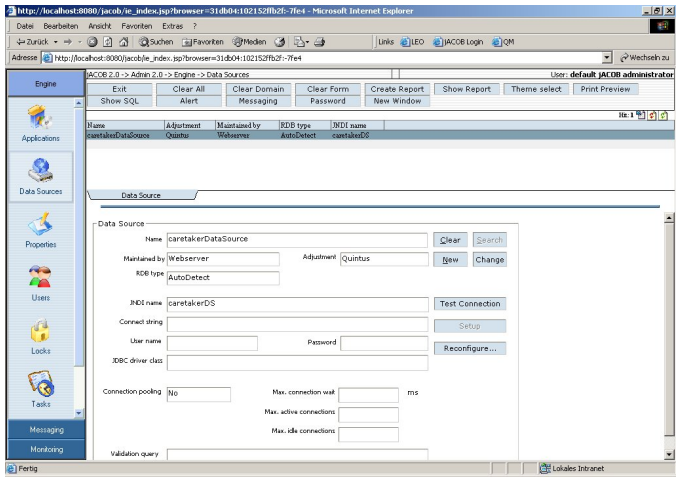
After a *JNDI data source* has been defined it will have the status defined with all jACOB™ applications using it.

Procedure

Perform the following steps to define a *JNDI data source*.

Note

A mouse click on a data source in the Assigned data sources table of the Applications form brings you directly to the Data sources form. In this case skip steps 1 to 3 and continue with step 4.

Step	Procedure
1	Open the Data Sources form of the Engine domain.
2	Press the button New.
3	Specify the Name of the data source to be defined.
4	Select Webserver in the Maintained by combo box.
5	Select AutoDetect in the RDB type combo box.
6	Specify the Adjustment.
7	Enter the JNDI name into the JNDI name text field.
8	Press the Save button. 
9	Press the button Test Connection to ensure that the jACOB™ application server can connect itself to the defined data source.

Defining a jACOB™ maintained data source without connection pooling

Purpose

This section describes how rarely accessed data sources that are shared between jACOB™ applications only are defined. If a data source is seldom accessed, it will not be necessary to set up a *connection pool* for this data source.

Impact

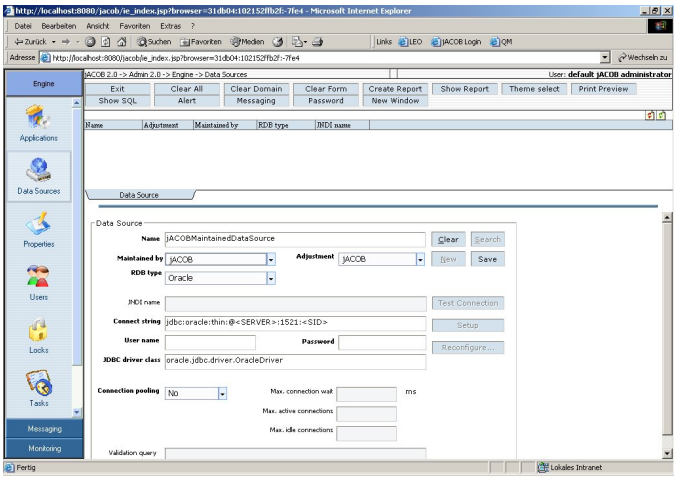
After a *jACOB™ maintained data source* has been defined, it will have the status defined within all jACOB™ applications using it.

Procedure

Perform the following steps to define a *jACOB™ maintained data source*.

Note

A mouse click on a data source in the Assigned data sources table of the Applications form brings you directly to the Data sources form. In this case skip steps 1 to 3 and continue with step 4.

Step	Procedure
1	Open the Data Sources form of the Engine domain.
2	Press the button New.
3	Specify the Name of the data source to be defined.
4	Select jACOB in the Maintained by combo box.
5	Select the database type of the data source in the RDB type combo box. The values of the Connect string and JDBC driver class are subject to special patterns. These depend on the specified RDB type. The jACOB™ Administration enters the patterns into these fields by default. 
6	Specify the Adjustment.
7	Enter the missing data into the pattern of the Connect string. For an Oracle data source the pattern is: "jdbc:oracle:thin:@<SERVER>:1521:<SID>". Replace <SERVER> with the host name or IP address of the Oracle database server. <SID> must be replaced with the Oracle SID (Service ID). For a MSSQL data source the pattern is: "jdbc:microsoft:sqlserver://<SERVER>:

Step	Procedure
	1433;DatabaseName=<DBNAME>;". Replace <SERVER> with the host name or IP address of the MSSQL database server. <DBNAME> must be replaced with the logical database name.
8	Specify the User name and Password of the owner/user of the data source.
9	Press the Save button.
10	Press the button Test Connection to ensure that the jACOB™ application server can connect itself to the defined data source.

Defining a jACOB™ maintained data source with connection pooling

Purpose

This section describes how frequently accessed data sources that are shared between jACOB™ applications only are defined. If a data source is accessed frequently, it will be more efficient to set up a *connection pool*.

For data sources accessed in a productive environment, it is recommended to set up a *connection pool*.

Impact

After the *jACOB™ maintained data source* with *connection pooling* has been defined it will have the status defined within all jACOB™ applications using it.

The *connection pool* will be set up by jACOB™ in accordance with the specifications entered into the Data sources form.

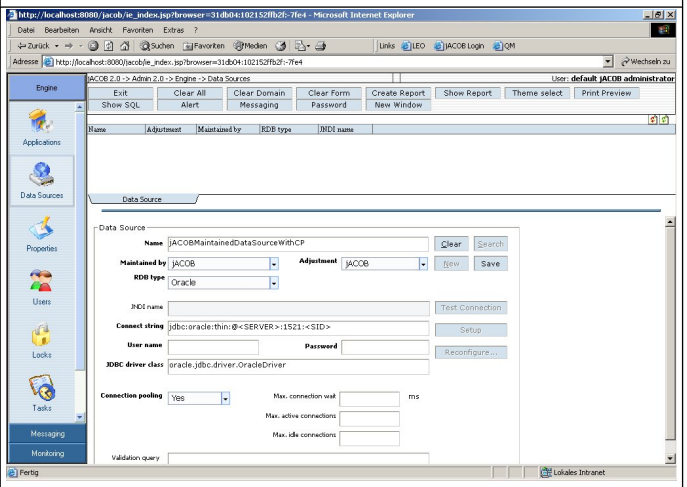
Procedure

Perform the following steps to define a *jACOB™ maintained data source* with *connection pooling*.

Note

A mouse click on a data source in the Assigned data sources table of the Applications form brings you directly to the Data sources form. In this case skip steps 1 to 3 and continue with step 4.

Step	Procedure
1	Open the Data Sources form of the Engine domain.
2	Press the button New.
3	Specify the Name of the data source to be set up.
4	Select jACOB in the Maintained by combo box.
5	Select the database type of the data source in the RDB type combo box. The values of the Connect string and JDBC driver class are subject to special patterns. These depend on the specified RDB type. The jACOB™ Administration enters the patterns into these fields by default.
6	Specify the Adjustment.
7	Specify the User name and Password of the owner/user of the data source and enter the missing data into the pattern of the Connect string. For an Oracle data source the pattern is: "jdbc:oracle:thin:@<SERVER>:1521:<SID>". Replace <SERVER> with the host name or IP address of the Oracle database server. <SID> must be replaced with the Oracle SID (Service ID). For a MSSQL data source the pattern is: "jdbc:microsoft:sqlserver://<SERVER>:1433;DatabaseName=<DBNAME>;". Replace <SERVER> with the host name or IP address of the MSSQL database server. <DBNAME> must be replaced with the logical database name.
8	Select Yes in the Connection pooling combo box. jACOB™ will enter default values depending on the RDB type into the *connection pooling* fields.

Step	Procedure
	
9	Specify the *connection pooling* data such as Max. connection wait, Max. active connections, Max. idle connections and the Validation query or accept the default values. Press the Save button.
10	Press the button Test Connection to ensure that the jACOB™ application server can connect itself to the defined data source.

Reconfiguring data sources

Purpose

The deployment of a new application version sometimes requires the reconfiguration of the data sources assigned to the application. This is the case if the structure of the data sources is not in line with the structure expected by the application.

The jACOB™ Administration provides a reconfiguration functionality to adapt the structure of a data source assigned to an application version. The following steps are performed:

- jACOB™ compares the structure of the data source with the structure expected by the application version.
- The SQL statements necessary for the adaptation are listed in a dialog window asking to confirm the execution of this procedure.
- After the confirmation jACOB™ performs the listed SQL statements.

  Caution

  Columns and tables that are no longer required are irreversibly dropped!
- Aborted SQL statements are listed in an extra dialog window, the Execution Summary window.

The Execution Summary window will appear only, if at least one SQL statement could not be executed successfully. In this case jACOB™ lists all SQL statements (unsuccessful and successful ones!) in this window. The data source will have to be migrated by hand accordingly.

E.g. a column of a table is set from NOT REQUIRED (the value NULL is valid!) to REQUIRED. The table contains data records with NULL in this column. These data records will have to be adapted by hand.

Impact

The structure of the data source is adapted to the structure expected by the application version.

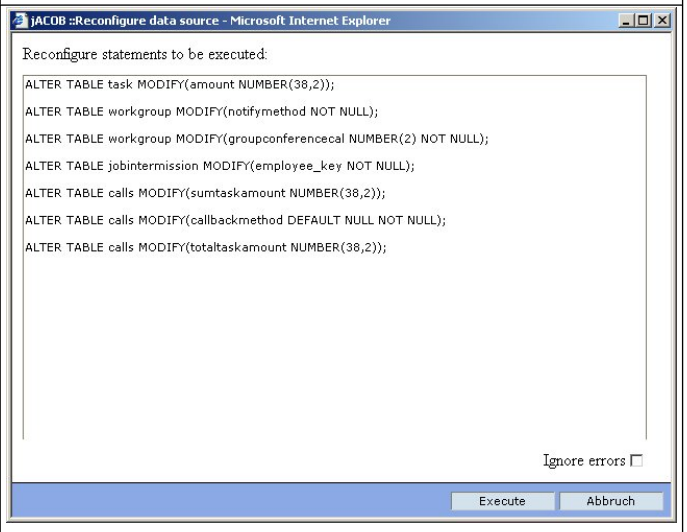
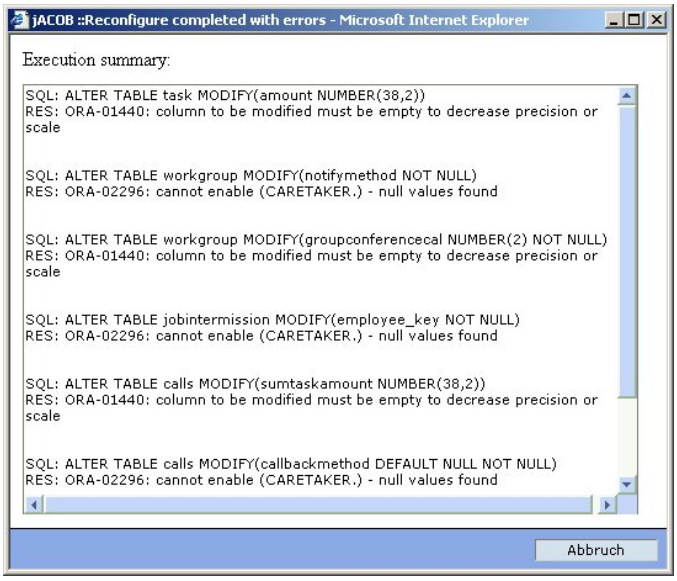
Caution

This procedure is not reversible. Columns and tables not required anymore are dropped! Therefore, the reconfiguration procedure has to be used carefully!

Procedure

Perform the following steps to reconfigure a data source to an application version:

Step	Procedure
1	Open the Data Sources form of the Engine domain.
2	Select the data source to be reconfigured and press the button Reconfigure....
3	Select the application version to which the data source should be reconfigured and confirm your choice. This action will probably take a while.
4	A dialog window will appear listing all SQL statements to be executed.

Step	Procedure
	 Enable the Ignore errors checkbox, if an interruption for each execution error is not wanted. At the end of the execution possible errors will be listed in the Execution Summary window. Press the Execute button to perform the reconfiguration.
5	If the reconfiguration has not been performed without any errors, the Execution Summary window will appear listing the errors found.  The affected tables will have to be adapted by hand.

Setting up data sources

Purpose

The jACOB™ Administration also enables the initial setup of data sources.

The setup procedure consists of four single steps:

1. The creation of a new data source schema, i.e. logical database, with a third party database tool,
2. The definition of the data source in the jACOB™ Administration,
3. The actual setup of the data source with the aid of the jACOB™ Administration and
4. The reconfiguration of the data source to an application version.

Note

This procedure enables the setup of jACOB™ data sources. Therefore, the Adjustment type in the Data Sources form must be set to jACOB.

Third party data sources have to be set up with the appropriate tools, e.g. Quintus data sources with the QDesigner.

Impact

The meta schema of the data source will be created according to the specified definitions.

Step 3, the actual setup, invokes the setup of database objects such as jACOB™ internal maintenance tables and *stored procedures*.

However, step 4, the reconfiguration, implies the setup of the database tables specific to the application.

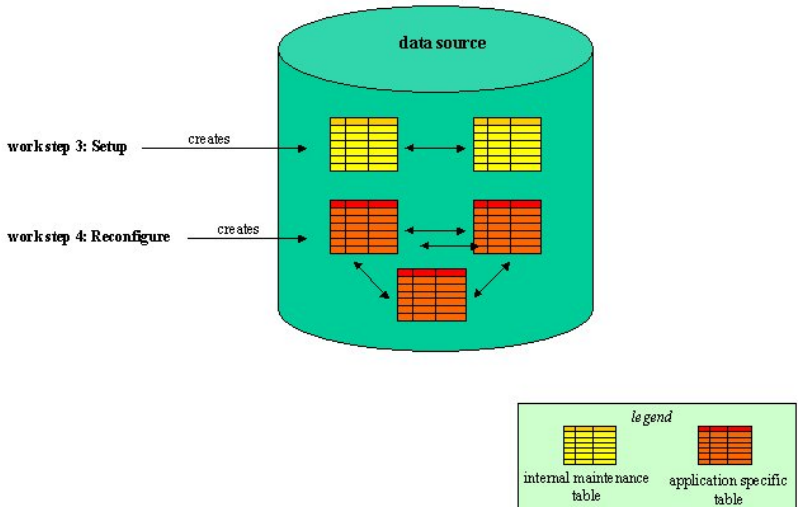


Figure 1.5. Setup of data source

Procedure

Perform the following steps to set up a data source:

Step	Procedure
1	Create a new data source schema, i.e. logical database with a third party database tool (please refer to the appropriate database documentation).

Step	Procedure
2	Define the data source with the Adjustment type jACOB in the Data Sources form as described earlier in this chapter.
3	Press the button Setup and confirm the action. This action could take a few moments.
4	Reconfigure the data source to the desired application version as described in the procedure Reconfiguring data sources.

Configuring the jACOB™ application server

Introduction	The jACOB™ application server can be configured to your requirements. This is done by means of predefined properties. More precisely, jACOB™ offers not only to configure the runtime in general but also to create application specific or even user specific adjustments.
jACOB properties	Configuration parameters specified for the application server in general are stored in jACOB properties.
jACOB application properties	jACOB application properties are configuration parameters specified for particular jACOB™ applications. The values of jACOB application properties override the values of *jACOB properties*.
User runtime properties	Configuration parameters for specific users of specific applications are stored in user runtime properties. The values of user runtime properties again override the values of *jACOB application properties*.
Priority hierarchy of properties	This leads to the following priority hierarchy of properties: 1. *jACOB properties* (lowest priority) 2. *jACOB application properties* 3. *user runtime properties* (highest priority) Example 1.1. Priority hierarchy of properties The Properties form (see below) shows the values of the property browser.common.max.records in the lower area. The browser.common.max.records property defines the common maximum of records displayed in browsers. The upper area of the Properties form shows the jACOB Property search browser displaying the result of an unconstrained search on the jACOB Property group. The *jACOB application property* value is set to 50 for the jACOB™ administration and overrides the *jACOB property* value of 10. Therefore, the jACOB Property search browser lists all 41 records found.
Form and domain	The configuration takes place within the Properties form of the Engine domain.

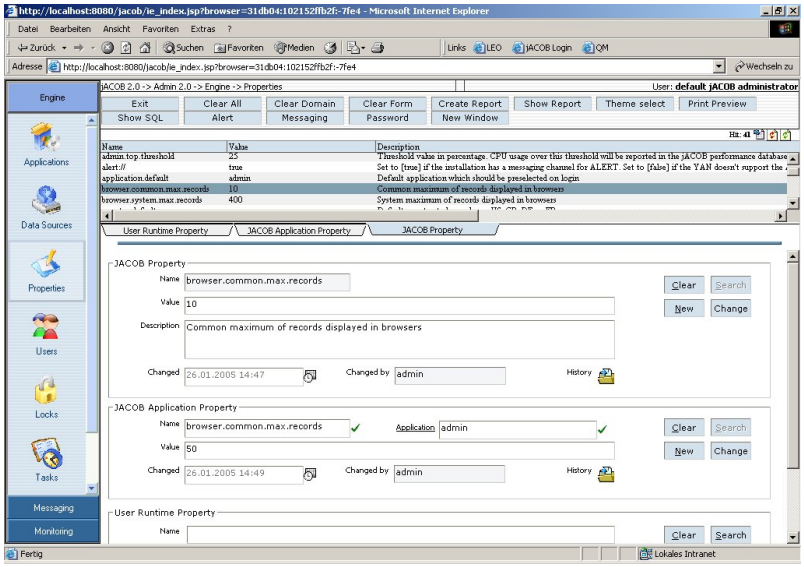


Figure 1.6. Properties form

The Properties form splits its information into the following groups:

- jACOB Property group
- jACOB Application Property group
- User Runtime Property group

jACOB Property group

The jACOB Property group is used to set and specify *jACOB™ properties*.

In addition to the Name, Value and Description the group shows the date and time the property was last Changed and Changed by which user.

Clicking the history icon  can retrieve a history of the selected property.

jACOB Application Property group

The jACOB Application Property group is used to set and specify *jACOB application properties*.

In addition to the Name and Value it must be specified which Application should be parameterized by the property.

As with *jACOB properties* a history is kept of all modifications executed on the properties.

User Runtime Property group

The User Runtime Property group is used to set and specify *user runtime properties*.

In addition to the Name and Value it must be specified which Application should be parameterized and for which user this property is valid.

Predefined properties

jACOB™ is delivered with a number of predefined properties. These properties will be described in the next sections.

The table below lists all predefined properties with their initial value. For some properties it is not possible to set an application specific or user specific value. This information is stated in the columns:

- *jP (jACOB Property)*
- *jAP (jACOB Application Property)*
- *URP (User Runtime Property)*

Table 1.1. Predefined properties

Name	Initial Value	jP	jAP	URP
Monitoring properties				
admin.memory.watch.interval	600	X		
admin.top.interval	120	X		
admin.top.threshold	25	X		
interval.check.alert	20	X	X	
interval.check.application	20	X	X	
interval.check.dialog	20	X	X	
interval.keepalive.application	10	X	X	
interval.keepalive.dialog	10	X	X	
interval.timeout.application	200	X	X	
interval.timeout.dialog	200	X	X	
sql.history.threshold	1000	X	X	
jAN properties				
alert://	true	X	X	X
email://	true	X	X	X
htmlemail://	true	X	X	X
rightfax://	true	X	X	X
sms://	true	X	X	X
Browser properties				
browser.common.max.records	100	X	X	X
browser.system.max.records	1000	X	X	X
gui.ffbrowser.inline	false	X	X	X
shortcut.foreignfield	113	X	X	X
Internationalization properties				
country.default		X	X	X
language.default		X	X	X
Toolbar properties				
ui.toolbar.alert	true	X	X	X
ui.toolbar.clearAll	true	X	X	X
ui.toolbar.clearFocus	true	X	X	X
ui.toolbar.clearForm	true	X	X	X
ui.toolbar.createReport	true	X	X	X
ui.toolbar.exit	true	X	X	X
ui.toolbar.messaging	true	X	X	X
ui.toolbar.newWindow	true	X	X	X
ui.toolbar.pretty	true	X	X	X
ui.toolbar.showReports	true	X	X	X
ui.toolbar.showSql	true	X	X	X
ui.toolbar.themeSelect	true	X	X	X
ui.toolbar.userPassword	true	X	X	X

Monitoring properties

Introduction

The monitoring properties configure the monitoring of the jACOB™ application server such as memory and CPU usage, the processing of SQL statements, etc.

The following monitoring properties exist and will be explained below:

- admin.memory.watch.interval
- admin.top.interval
- admin.top.threshold
- sql.history.threshold
- interval.check.alert
- interval.timeout.application
- interval.check.application
- interval.keepalive.application
- interval.timeout.dialog
- interval.check.dialog
- interval.keepalive.dialog

admin.memory.watch.interval

The property admin.memory.watch.interval specifies an interval in seconds. The jACOB™ application server inspects the memory usage of the Java Virtual Machine at the end of this interval. At this point in time it also calls the *garbage collector*. The memory usage is logged and can be viewed in the System form of the Monitoring domain.

Note

This property will only have an effect, if the task MemoryWatch has been activated.

To avoid a slowdown of the application server, the admin.memory.watch.interval should be set to a value greater than 30 seconds.

admin.top.interval and admin.top.threshold

The property admin.top.interval also defines an interval in seconds. However, the jACOB™ application server inspects the CPU usage at the end of this interval.

If the CPU usage exceeds the threshold defined by the property admin.top.threshold, this will be logged and can be also viewed in the System form of the Monitoring domain. The admin.top.threshold is specified in percentage of CPU usage.

Note

To avoid a slowdown of the application server, the admin.top.interval should be set to a value greater than 30 seconds.

These properties will only have an effect, if the task Top has been activated. Due to the fact that the analysis of the CPU usage is specific to the running operating system a clean processing of the task Top is guaranteed only for the following operating systems: Solaris, Linux and Windows.

sql.history.threshold	The property sql.history.threshold specifies the maximum execution time in milliseconds to process an SQL statement. If the execution takes longer than sql.history.threshold, the SQL statement will be logged to the history database and can be viewed in the SQL form of the Monitoring domain.
interval.check.alert	Also the property interval.check.alert specifies an interval in seconds. At the end of this interval the jACOB™ application server sends a request to the corresponding application for each *user session* asking for new alert messages for the current session. New alert messages will be displayed in a popup window on the client side.
interval.timeout. application, interval.check. application and interval. keepalive.application	The properties interval.timeout.application, interval.check.application and interval.keepalive.application all define intervals in seconds. Due to the fact that HTTP is a stateless protocol, these properties are required to check whether an application window is still open. The jACOB™ application server creates a timer for each application window opened on the client side. The timer's timeout is set to interval.timeout.application. The application server checks every interval.check.application seconds whether the timer has expired. If the timer has expired, all resources allocated to this application window will be released. At every interval.keepalive.application seconds the client sends a ping message to the application server to inform the server that it is still alive. Hereafter, the server resets the timer for the application window. This affects the following: • interval.keepalive.application < interval.timeout.application • interval.check.application < interval.timeout.application **Note** To avoid a slowdown of the application server, the interval.keepalive.application should be set to a value greater than 60 seconds.
interval.timeout.dialog, interval.check.dialog and interval.keepalive. dialog	The same mechanism exists for dialogue windows. The properties interval.timeout.dialog, interval.check.dialog and interval.keepalive.dialog define the intervals for controlling the lifetime of a dialogue window.

jAN properties

Introduction

jAN is the abbreviation for jACOB™ Application Notifier, a message router for alerts, faxes, SMS messages and emails. For a detailed description of the *jAN* please refer to the jAN system manual.

jAN properties specify which messaging channels of the *jAN* are supported or should be supported for e.g. a certain user or application.

The following jAN properties exist and will be explained below:

- alert://
- email://
- htmlemail://
- rightfax://
- sms://

alert://	The property alert:// determines whether a messaging channel for alerts is supported by the *jAN* or should be supported for e.g. a certain user or application.
email://	The property email:// specifies whether a messaging channel for emails is supported by the *jAN* or should be supported for e.g. a certain user or application.
htmlemail://	The property htmlemail:// defines whether a messaging channel for HTML-emails, i.e. emails with HTML content, is supported by the *jAN* or should be supported for e.g. a certain user or application.
rightfax://	The property rightfax:// constitutes whether a messaging channel for faxes is supported by the *jAN* or should be supported for e.g. a certain user or application.
sms://	The property sms:// determines whether a messaging channel for SMS messages is supported by the *jAN* or should be supported for e.g. a certain user or application.

Browser properties

Introduction

Browser properties configure the jACOB™ browsers: search browser, inform browser and foreign field browser.

The following browser properties exist and will be explained below:

- browser.common.max.records
- browser.system.max.records
- gui.ffbrowser.inline
- shortcut.foreignfield

browser.common.max.records

The property browser.common.max.records specifies the common maximum number of records displayed in search browsers and inform browsers.

This maximum is used to limit the displayed records found by an ordinary search. This is initiated by pressing the Search button or clicking the Repeat last search icon  in the upper right corner of the browser.

browser.system.max.records

However, the property browser.system.max.records defines the system's maximum number of records displayed in search browsers.

This maximum is used to limit the number of displayed records found by an extended search. An extended search is initiated by clicking the Last search with system maximum icon  in the upper right corner of the search browser.

Note

To avoid long response times, the values of browser.common.max.records and browser.system.max.records should be set as low as possible. In particular for clients with access to the server via small bandwidth connections it is recommended to set the values lower than the default values.

gui.ffbrowser.inline

The property gui.ffbrowser.inline defines the presentation of foreign field browsers.

If the value is set to "true", foreign field browsers will be presented in inline windows. If the value set to "false", foreign field browsers will appear in external windows. The presentation in inline windows is faster than the presentation in external windows. However, inline windows cannot be moved.

shortcut.foreignfield

A shortcut for the search on foreign fields can be declared by setting the value of the property shortcut.foreignfield. The shortcut replaces a click on the magnifier icon  at the right hand side of foreign fields.

Internationalization properties

Introduction	Language settings are configured by the internationalization properties. In jACOB™ language settings have the following format: <Language>_[<Country>]
language.default	The <Language> is defined by the property language.default. If no language is specified, i.e. the value of language.default is empty, jACOB™ will use the language setting of the internet browser used. Currently the following languages are supported: • en for English • de for German
country.default	The <Country> is optional and can be specified by the property country.default. It defines the country specific acuteness of a language, e.g. en_US and en_GB.

Toolbar properties

Introduction

Toolbar properties configure the toolbar displayed in the upper part of jACOB™ application windows.

Figure 1.7. Toolbar buttons

Available toolbar properties

The toolbar consists of a number of buttons. These buttons can be activated/deactivated by setting the appropriate properties.

The following table lists the available toolbar buttons and their corresponding properties:

Table 1.2. Toolbar properties

Toolbar button	Property
Alert	ui.toolbar.alert
Clear All	ui.toolbar.clearAll
Clear Domain	ui.toolbar.clearFocus
Clear Form	ui.toolbar.clearForm
Create Report	ui.toolbar.createReport
Exit	ui.toolbar.exit
Messaging	ui.toolbar.messaging
New Window	ui.toolbar.newWindow
Pretty Print	ui.toolbar.pretty
Show Reports	ui.toolbar.showReports
Show SQL	ui.toolbar.showSql
Themes	ui.toolbar.themeSelect
Password	ui.toolbar.userPassword

Miscellaneous properties

Miscellaneous properties

Following miscellaneous properties are predefined and will be explained below:

- application.default
- gui.debug
- gui.window.prefix
- user.theme.default

application.default

The value of the application.default property determines which jACOB™ application should appear by default in the application field of the login screen. Please note, that the name of the application only should be entered for this property.

Example 1.2. Property application.default

The figure below shows the effect of setting the value of application.default to "admin".

gui.debug

The property gui.debug is for internal use only and therefore, will not be further explained.

gui.window.prefix

With the aid of the property gui.window.prefix the prefix in titles of dialogue windows and application windows can be defined.

Example 1.3. Property gui.window.prefix

The figure below shows the effect of setting the value of gui.window.prefix to "jACOB::".

user.theme.default

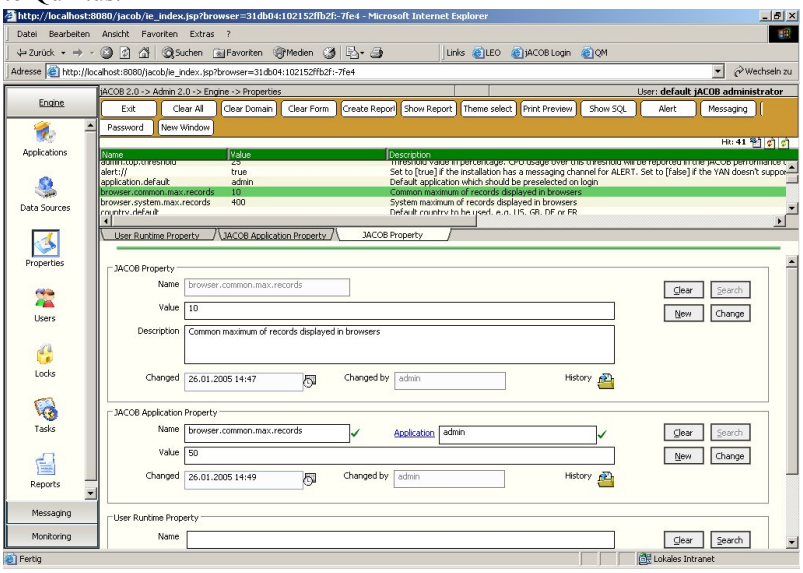
The value of the property user.theme.default determines the look&feel of jACOB™ applications.

The following themes currently exist:

- Apple
- Quintus
- QuintusBold
- blueXP
- custinfo
- Standard
- iTheme
- tarragon

Example 1.4. Property user.theme.default

The figure below shows the effect of setting the value of user.theme.default to Quintus.



Administrating user sessions and jACOB™ administrator accounts

Introduction

Before running a jACOB™ application the user has to authenticate against the application. Hereafter, jACOB™ opens a *user session*.

The jACOB™ Administration offers a service to administrate these *user sessions*. With the aid of this service you are able overview all currently active *user sessions* and terminate a *user session*, if necessary.

Even jACOB™ administrators have to authenticate against the jACOB™ Administration. The service also administrates their *user sessions* and furthermore enables the creation of new jACOB™ administrator accounts.

Form and domain

The administration of user sessions and jACOB™ administrator accounts takes place within the Users form of the Engine domain.

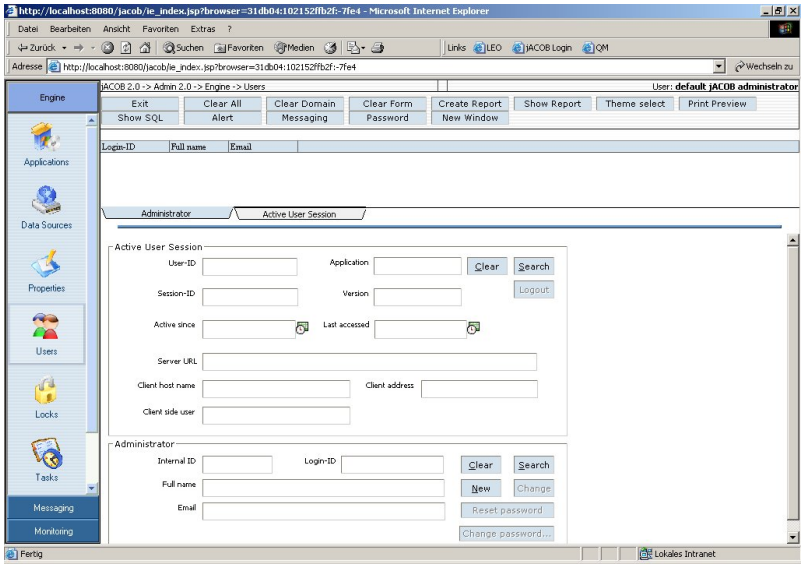


Figure 1.8. Users form

The information is devided into the following groups:

- Active user session group
- Administrator group

The Active user session group shows the following information of a *user session*:

- User-ID of the user
- Name of the running Application
- Session-ID
- Version of the running application
- Date and time the *user session* is Active since

- Date and time the *user session* Last accessed the running application
- Client host name
- Client address
- Client user name

The Administrator group shows the following information of a jACOB™ administrator account:

- Unique, not modifiable Internal-ID assigned once to the administrator
- Unique but modifiable Login-ID
- Full name
- Email address

Procedures

The following procedures will be described in the next sections:

- Retrieving currently active user sessions
- Terminating user sessions
- Creating new jACOB™ administrator accounts
- Changing the password of a jACOB™ administrator
- Resetting the password of a jACOB™ administrator

Retrieving currently active user sessions

Purpose

This section describes how an overview of all currently active *user sessions* matching certain search criteria can be retrieved from jACOB™.

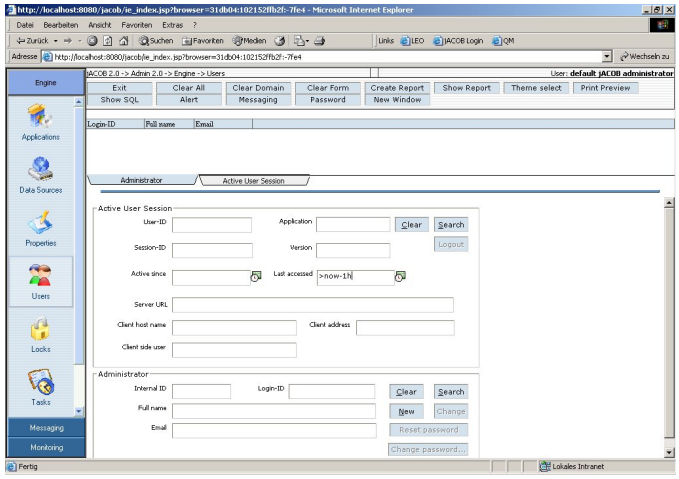
Impact

If only one *user session* is matching the search criteria, the data of this *user session* will be *backfilled* into the Active user session group of the Users form. Furthermore, the *user session's* application data will be *backfilled* into the Installed application group of the Applications form.

Procedure

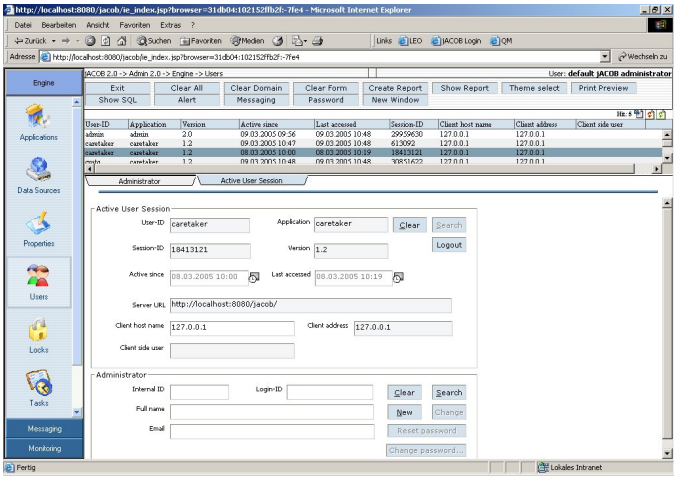
Perform the following steps to retrieve an overview of all currently active *user sessions* matching certain search criteria.

The initial situation is an empty domain.

Step	Procedure
1	Open the Users form of the Engine domain.
2	Enter the search criteria into the fields of the Active user session group. If you want to search e.g. all user sessions having accessed jACOB™ in the last hour, enter ">now-1h" into the Last accessed field. 
3	Press the Search button in the Active user session group.

Terminating user sessions

Purpose	This section describes how an active *user session* can be terminated.
Impact	The specified *user session* will be terminated immediately. After the next request, the user will see the jACOB™ login page with a message informing the user of the termination. Unsaved modifications will be lost!
Procedure	Perform the following steps to terminate an *user session*:

Step	Procedure
1	Search and select the *user session* to be deleted as described in Retrieving currently active user sessions. 
2	Press the Terminate... button and confirm the query that follows.

Creating new jACOB™ administrator accounts

Purpose

Initially, the jACOB™ Administration already provides one administrator account. This administrator account has the Login-ID "admin". In addition the jACOB™ Administration offers the possibility to create further administrator accounts.

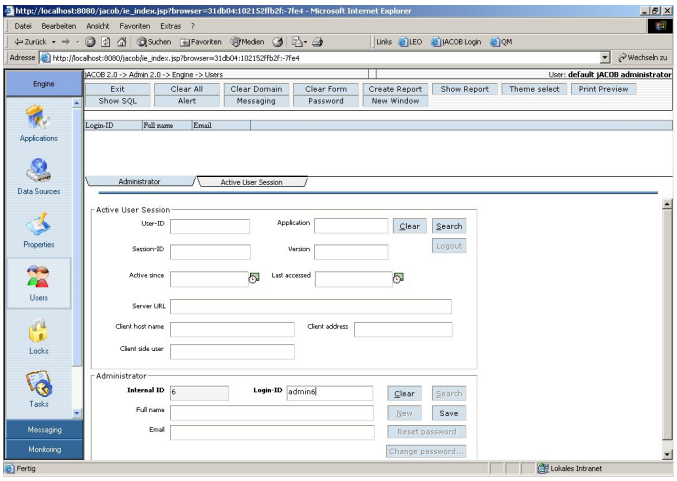
This section describes how a new jACOB™ administrator account is created.

Impact

The jACOB™ Administration will create a new administrator account with an initially empty password.

Procedure

Perform the following steps to create a new jACOB™ administrator account:

Step	Procedure
1	Press the New button in the Active user session group.
2	Specify a unique Login-ID for the new administrator account. 
3	Enter the Full name and Email address, if desired.
4	Press the Save button.

Changing the password of a jACOB™ administrator

Purpose

jACOB™ administrators have to authenticate against the jACOB™ Administration. Initially, their password is an empty string. One of the first actions after the creation of an administrator account will probably be to change the password.

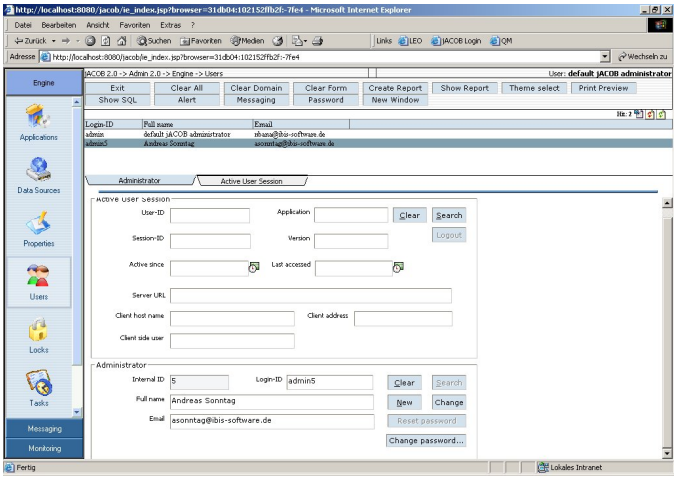
This section describes how the password of a jACOB™ administrator can be changed.

Impact

The jACOB™ Administration will change the password. From then on the administrator will have to authenticate against the jACOB™ Administration with the new password.

Procedure

Perform the following steps to change a jACOB™ administrator's password:

Step	Procedure
1	Select the administrator account concerned in the Administrator group of the Users form. 
2	Press the Change password... button. A dialog window will appear prompting to enter the new password.
3	Enter the new password into the text field of the dialog window and confirm the change.

Resetting the password of a jACOB™ administrator

Purpose

The password of a jACOB™ administrator may also be reset e.g. if the administrator has forgotten the password.

This section describes how the password of a jACOB™ administrator can be reset.

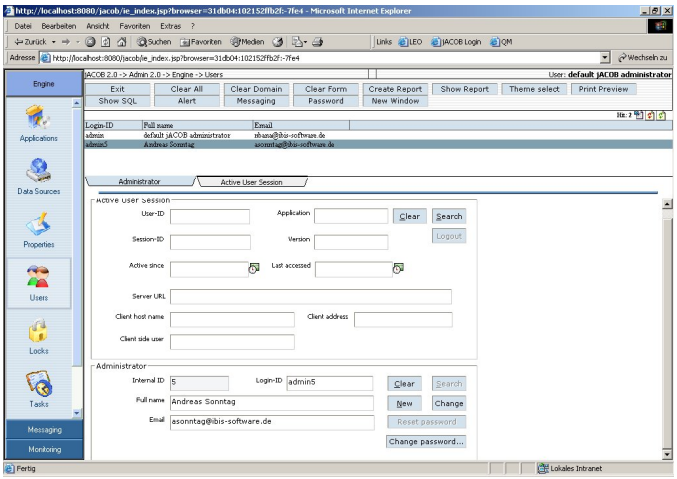
Impact

The jACOB™ Administration will set the password to an empty string. The administrator concerned will have to authenticate against the jACOB™ Administration with an empty string at the next login.

To prevent malpractice of any third party the administrator should change the password immediately as described in Changing the password of a jACOB™ administrator.

Procedure

Perform the following steps to reset a jACOB™ administrator's password:

Step	Procedure
1	Select the administrator account concerned in the Administrator group of the Users form. 
2	Press the Reset password button.

Administrating locks on data records

Introduction

jACOB™ uses the *pessimistic locking* technique to lock data records. I.e. as soon as a user presses the Change button the data record concerned is locked and stays in this mode until the user presses the Save button or any Clear button respectively, i.e. Clear, Clear All, Clear Domain or Clear Form.

Form and domain

Record locks can be viewed and removed with the aid of the Locks form of the Engine domain.

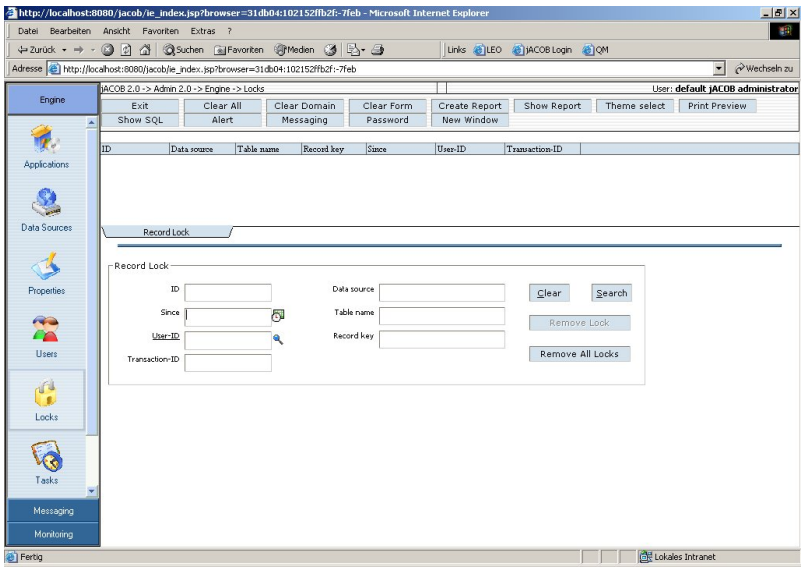


Figure 1.9. Locks form

A record lock has the following data:

- ID of the record lock
- Data source, Table name and Record key of the locked record
- Locked Since date and time
- User-ID and Transaction-ID the record lock was performed by

Procedures

The following procedures will be described in the next sections:

- Removing a record lock

Removing a record lock

Purpose

As stated above, a locked record stays locked until the user presses the Save button or any Clear button respectively. This means that a record can stay locked at any given time. This is sometimes not desired, e.g. a user starts to edit a record and leaves his desk before saving the record. Thus a jACOB™ administrator is sometimes asked explicitly to remove a record lock.

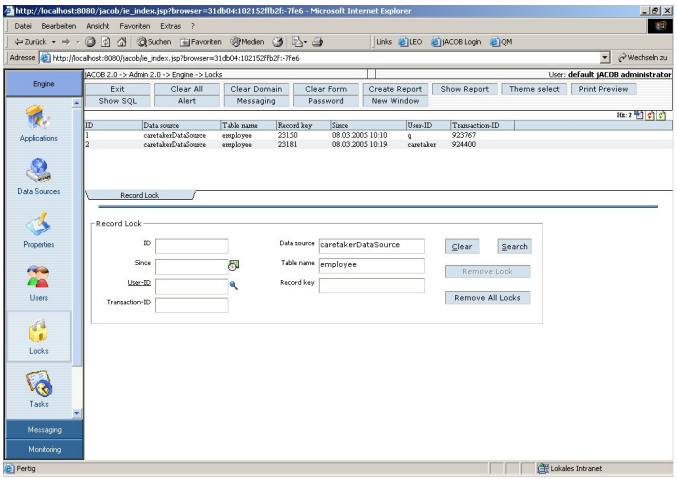
The following scenario describes how a record lock is removed.

Impact

The record lock is removed and the record can be edited by another user.

Procedure

Perform the following steps to remove a record lock:

Step	Procedure
1	Open the Locks form of the Engine domain.
2	Enter the search criteria into the appropriate fields, e.g. the name of the particular data source and table, and press the Search button. 
3	Select the particular record and press the Remove Lock button. **Note** Pressing the Remove All Locks button removes all locks listed in the search browser Record Lock.

Administrating jACOB tasks

Introduction

jACOB™ enables the handling of multiple jACOB tasks. A jACOB task is a background job executed periodically within the jACOB™ application server.

Each *jACOB task* is assigned to a scheduler. The scheduler periodically executes all tasks assigned sequentially. jACOB™ represents schedulers as threads.

jACOB application tasks and jACOB internal tasks

Most *jACOB tasks* are assigned to a particular application version. Application programmers can define their own tasks. These tasks are called jACOB application tasks.

There are also jACOB internal tasks. These are predefined tasks e.g. needed to maintain user sessions.

Form and domain

jACOB tasks are administrated within the Tasks form of the Engine domain.

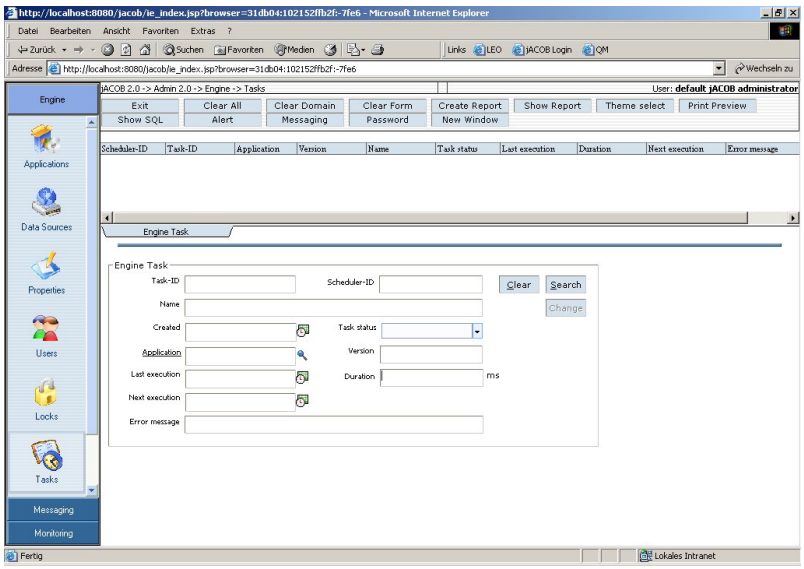


Figure 1.10. Tasks form

A *jACOB task* has the following data structure:

- Task-ID
- Scheduler-ID
- Name of the *jACOB task*
- Date and time the *jACOB task* has been Created
- Task status
- Application and Version the *jACOB task* is assigned to
- Last execution date and time and its Duration in milliseconds

- Next execution date and time
- Eventual Error message in case of erroneous Last execution

Task status

The status of a *jACOB task* can be:

- scheduled or
- hibernated

Caution

The Task status can be modified in the Tasks form. However, to guarantee a faultless working of the jACOB™ application server it is highly recommended to only modify the task status of *jACOB application tasks*.

Next execution

The Next execution field displays the time of the next planned task execution. The Next execution is also displayed for hibernated *jACOB tasks*. In this case Next execution will indicate the point in time the task will be executed, if the Task status is set to scheduled.

Procedures

The following procedures will be described in the next sections:

- Viewing jACOB tasks with an incorrect last execution

Viewing jACOB tasks with an incorrect last execution

Purpose

Errors might occur during the execution of a *jACOB task*.

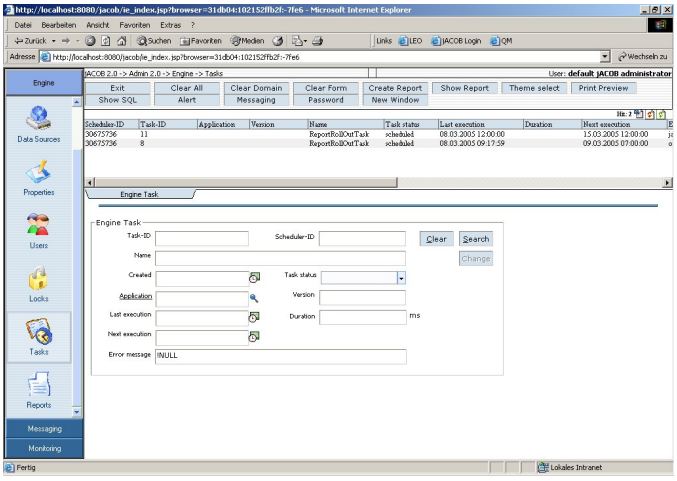
The following scenario describes how to list all *jACOB tasks* that performed incorrectly at their last execution.

Impact

The erroneous task executions are listed and the jACOB™ administrator can analyze the potential problem.

Procedure

Perform the following steps to list all *jACOB tasks* that showed errors at their last execution:

Step	Procedure
1	Open the Tasks form of the Engine domain.
2	Enter "!NULL" into the Error message text field.
3	Press the Search button. 

Administrating reports

Introduction

jACOB™ offers the possibility to create reports from every jACOB™ application.

In addition users can subscribe to reports so that these will be sent to them daily or weekly at a certain time.

Internally, jACOB™ generates a report definition in a jACOB™ specific XML format for each report.

The reports and their corresponding report definitions can be viewed and modified in the jACOB™ Administration.

Form and domain

Reports are administrated within the Reports form of the Engine domain.

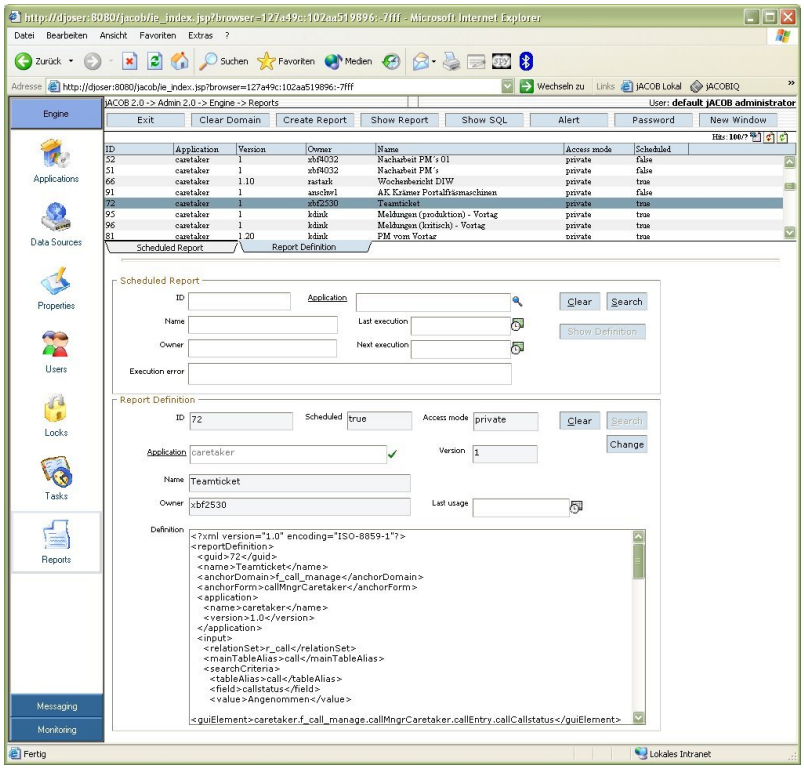


Figure 1.11. Reports form

The information is devided into the following groups:

- Scheduled Report
- Report Definition

Scheduled reports

A scheduled report is a report that someone has subscribed to. Such a report will be scheduled to be sent daily or weekly at a certain defined time.

The group Scheduled Report displays the data structure of a scheduled report:

	• ID of the report • Application the report is assigned to • Name of the report • Date and time of the Last execution and the Next execution • Owner of the report • Execution error message, if the last execution has failed
Report definitions	The group Report Definition displays the data structure of a jACOB™ report definition: • ID of the report • Scheduled state • Access mode • Application and Version the report is assigned to • Name of the report • Owner of the report • Last usage of the report • XML Definition of the report
Scheduled state	The Scheduled state shows whether a subscription to the report exists. If a subscription exists, the report's Scheduled state will be set to true.
Access mode	The Access mode of a report can be: • private, i.e. the report is only visible to the owner of the report; • public, i.e. the report is visible to all users; or • shared.
Procedures	The following procedures will be described in the next sections: • Viewing all reports scheduled by a particular user

Viewing all reports scheduled by a particular user

Purpose

A user complains that he/she does not receive the report he/she has subscribed to.

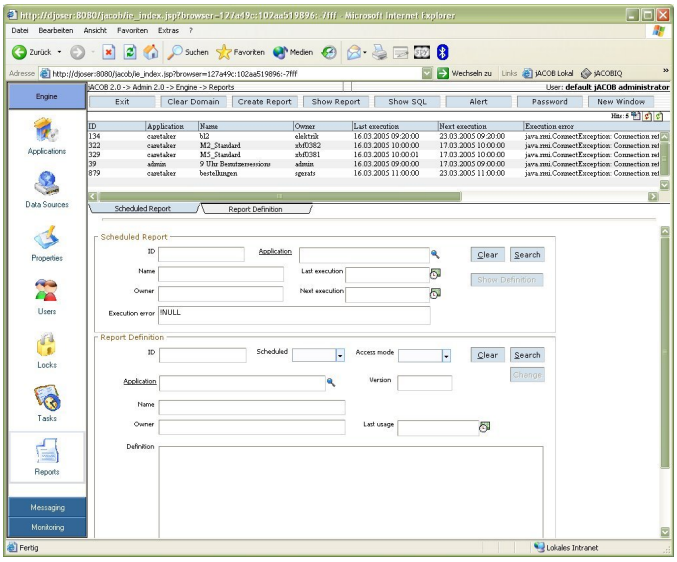
Scheduled reports that were not performed correctly at their last execution, can be viewed with the aid of a constrained search in the Scheduled Report group of the Reports form.

Impact

All reports scheduled by the user that were not performed correctly at their last execution, are listed in the Scheduled Report search browser. The administrator can view the appropriate report and analyze the problem.

Procedure

Perform the following steps to view all reports scheduled by a particular user, that were not executed correctly at their last execution:

Step	Procedure
1	Open the Reports form of the Engine domain.
2	Enter the name of the user into the Owner field of the Scheduled Report group.
3	Enter "!NULL" into the Execution error field. 
4	Press the Search button.
5	Click on the respective entries of the search browser to inspect the reports one after another.

Licensing the jACOB™ application server

Introduction	Running the jACOB™ application server requires the registration with a valid license key.
jACOB™ license keys	A jACOB™ license key is characterized by: • Its period of validity and • Its maximum number of concurrent *user sessions*.
Registration with a license key	During the first connection to the jACOB™ application server the user is asked to enter a valid license key. When this license key has expired or the maximum number of concurrent *user sessions* turns out to be too low, it will be necessary to install another license key. **Note** A jACOB™ administrator can always log into the jACOB™ Administration. This enables the installation of a new license key when the current key has expired or allows Terminating user sessions when the maximum number of concurrent *user sessions* has been reached.
Form and domain	New license keys are installed with the aid of the Applications form of the Engine domain.
Procedures	The following procedures will be described in the next sections: • Installing a new license key

Installing a new license key

Purpose

When a license key has expired or the maximum number of concurrent *user sessions* turns out to be too low, it will be necessary to install a new license key. The following scenario shows how to install a new license key.

Impact

The period of validity and the maximum number of concurrent *user sessions* will be set in accordance with the license key.

Procedure

Perform the following steps to install a new license key:

Step	Procedure
1	Open the Applications form of the Engine domain.
2	Press the button Enter jACOB license key.... After this a dialog box will pop up asking to enter the license key.
3	Enter the license key e.g. with the aid of cut&paste and press the Save button. 
4	Press the button Show jACOB license... for control. 

Chapter 2. The Messaging domain

Introduction

The Messaging domain is the starting point for the administration of messages from jACOB™ applications.

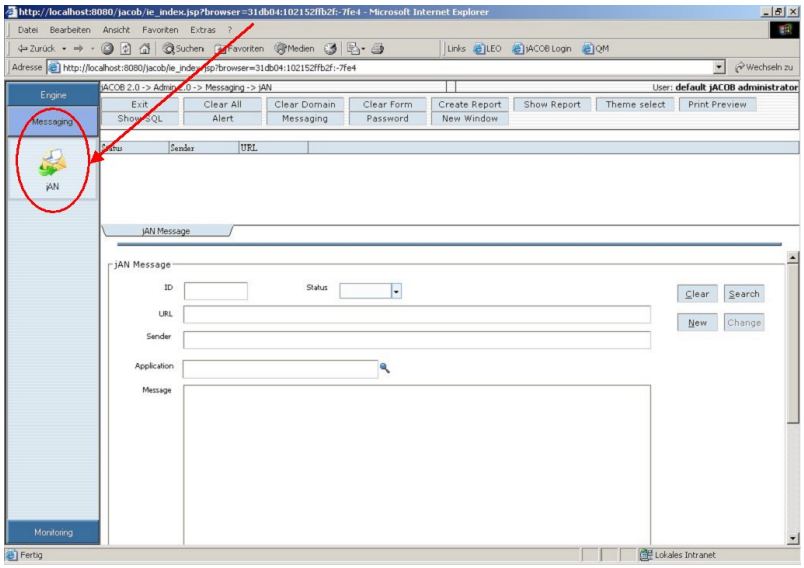


Figure 2.1. Messaging domain

What takes place in the Messaging domain?

The following actions take place in the Messaging domain:

- Administrating messages

Administrating messages

Introduction

jACOB™ offers the possibility to send messages from every jACOB™ application. jACOB™ stores these messages in a message queue. The external message router *jAN* subsequently processes the messages from this message queue.

Form and domain

The message queue can be viewed within the jAN form of the Messaging domain.

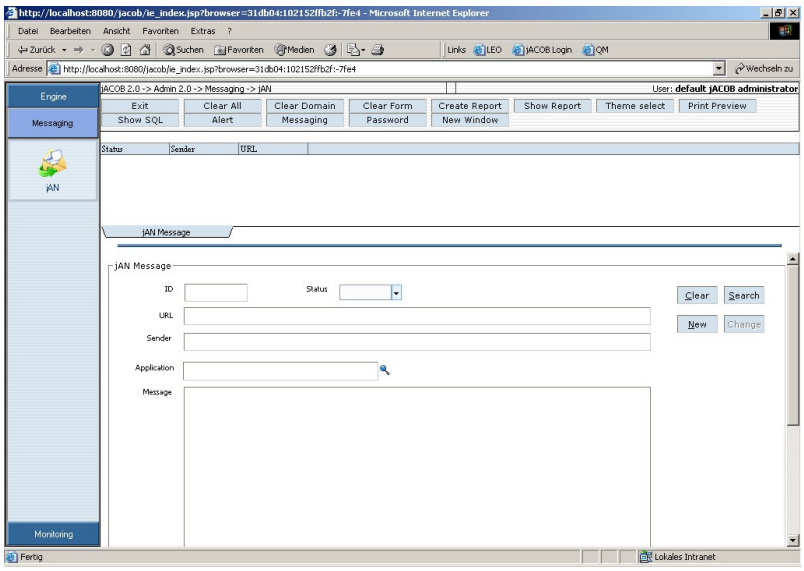


Figure 2.2. jAN form

The jAN form displays the data structure of a *jAN* message:

- ID of the message
- Status of the message
- Creation date of the message
- URL of the message recipient
- Sender of the message
- The Application where the message has been generated from
- Message text
- Error message text

Status of a message

The Status of a message can be:

- New or
- Error

	The initial Status of a message is New. Messages that could not be processed by the *jAN* obtain the Status Error. Details about the error can be found in the Error message text field.
Procedures	The following procedures will be described in the next sections: • Viewing messages with an error message status

Viewing messages with an error message status

Purpose

The message router *jAN* processes the messages from the message queue. Messages that could not be processed by the *jAN* obtain the Status Error.

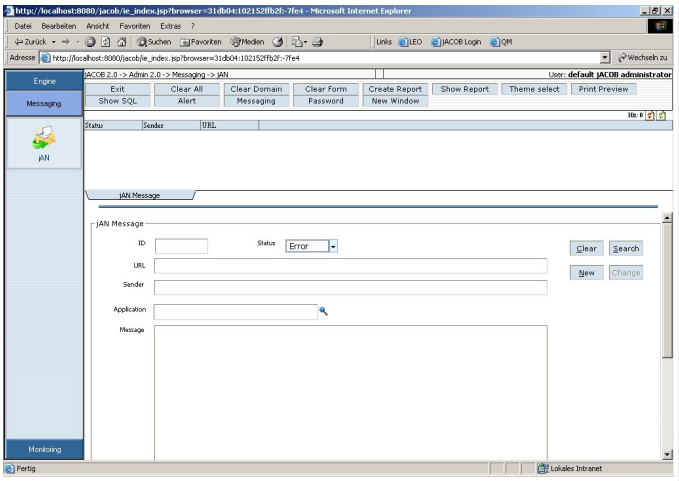
Messages with Status Error can be viewed and analyzed with the aid of a constrained search.

Impact

All messages with Status Error are listed in the search browser. With the aid of the Error message the administrator can analyze the reasons for these problems, e.g. *jAN* not correctly configured, email server unavailable, fax server unavailable, etc.

Procedure

Perform the following steps to view messages with an error message status:

Step	Procedure
1	Open the jAN form of the Messaging domain.
2	Choose Error in the Status field.
3	Press the Search button. 
4	Click sequentially on the respective entries of the search browser to analyze potential problems.

Chapter 3. The Monitoring domain

Introduction

The Monitoring domain is the starting point for the monitoring of SQL data sources and system resources.

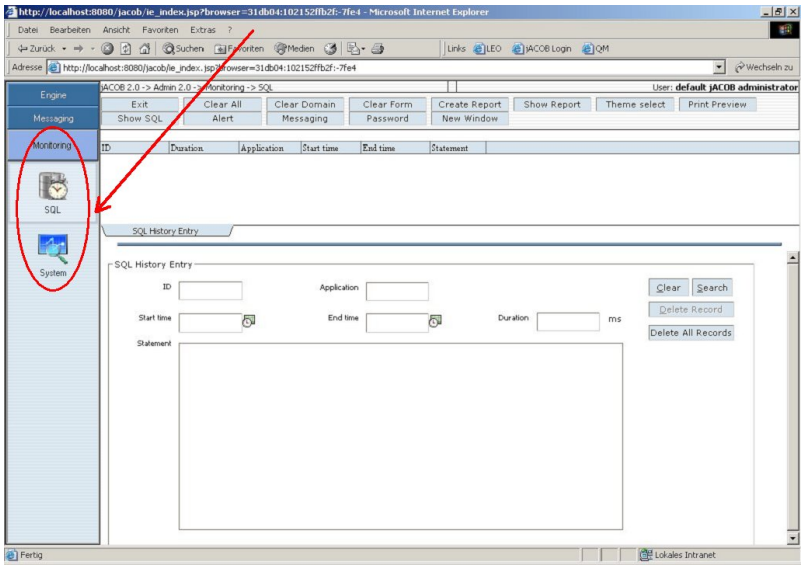


Figure 3.1. Monitoring domain

What takes place in the Monitoring domain?

The following actions take place in the Monitoring domain:

- Monitoring SQL data sources
- Monitoring system resources

Monitoring SQL data sources

Introduction

jACOB™ monitors all requests to SQL data sources. Thus time consuming data source requests can be detected in a simple way.

As already mentioned in the section Monitoring properties the property sql.history.threshold configures the monitoring of SQL data sources. If the duration of a SQL statement exceeds this threshold, this will be logged in the SQL history database.

Form and domain

The SQL history database can be viewed and emptied with the aid of the SQL form of the Monitoring domain.

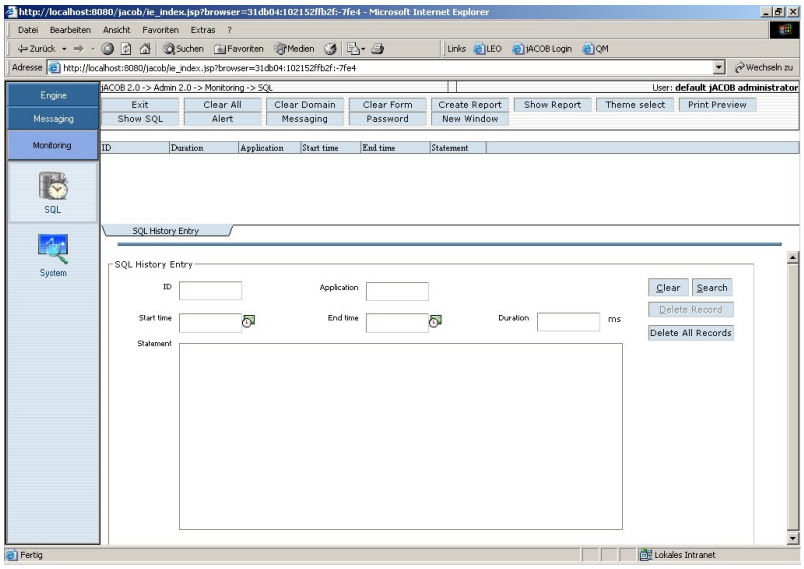


Figure 3.2. SQL form

The SQL form displays the data structure of an entry in the SQL history database:

- Internal ID of the entry
- Data source the request was made on
- The Application the SQL statement was released from
- Start time, End time and resulting Duration in milliseconds
- SQL Statement itself

Emptying of the SQL history database

The jACOB™ administrators are solely responsible for the regular emptying of the SQL history database. This can be done by pressing the Delete All Records button.

Procedures

The following procedures will be described in the next sections:

- Viewing the time consuming SQL statements of an application

Viewing the time consuming SQL statements of an application

Purpose

Time consuming SQL statements can slow down the processing of an application. Therefore, it is important to detect and analyze these SQL statements in order to perform a subsequent data source optimization.

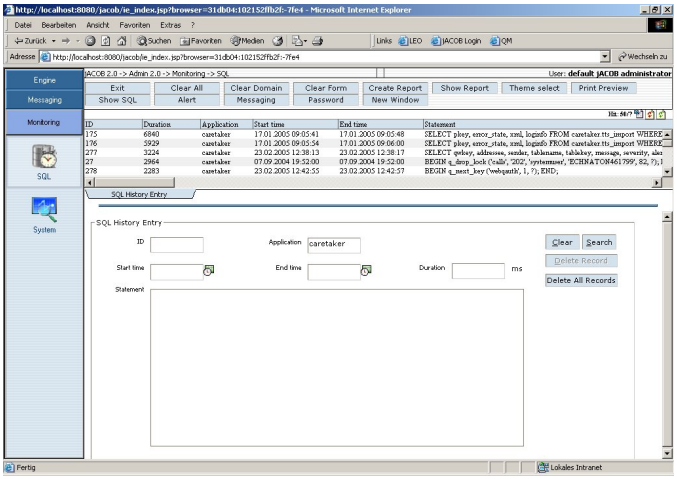
SQL statements slowing down a particular application can be viewed with the aid of a constrained search on the SQL history database.

Impact

All SQL statements that slowed down the application are listed in the search browser. The administrator can analyze the problems and perform an optimization of the data sources.

Procedure

Perform the following steps to view the time consuming SQL statements of an application:

Step	Procedure
1	Open the SQL form of the Monitoring domain.
2	Enter the name of the application into the Application field.
3	Press the Search button. 
4	Click on the respective entries of the search browser one after another to analyze potential problems.

Monitoring system resources

Introduction

jACOB™ monitors the system resources main memory and CPU. This simplifies to optimize the distribution of the system resources and prevents shortages of system resources.

Memory monitoring

The jACOB™ task MemoryWatch is responsible for the monitoring of the main memory.

As already mentioned in the section Monitoring properties the property admin.memory.watch.interval configures the monitoring of the main memory. jACOB™ inspects the memory usage of the Java Virtual Machine at the end of this interval, runs the *garbage collector* and inspects again the memory usage. The evaluation of the memory usage before and after the garbage collection is logged in the system-monitoring database .

CPU monitoring

The jACOB™ task Top is responsible for monitoring the CPU.

As already stated in the section Monitoring properties the properties admin.top.interval and admin.top.threshold configure the monitoring of the CPU. jACOB™ inspects the CPU usage at the end of this interval. If the CPU usage exceeds the defined threshold, jACOB™ will make a snapshot of the processor load and store it in the system monitoring database.

Form and domain

The system monitoring database can be viewed and emptied with the aid of the System form of the Monitoring domain.

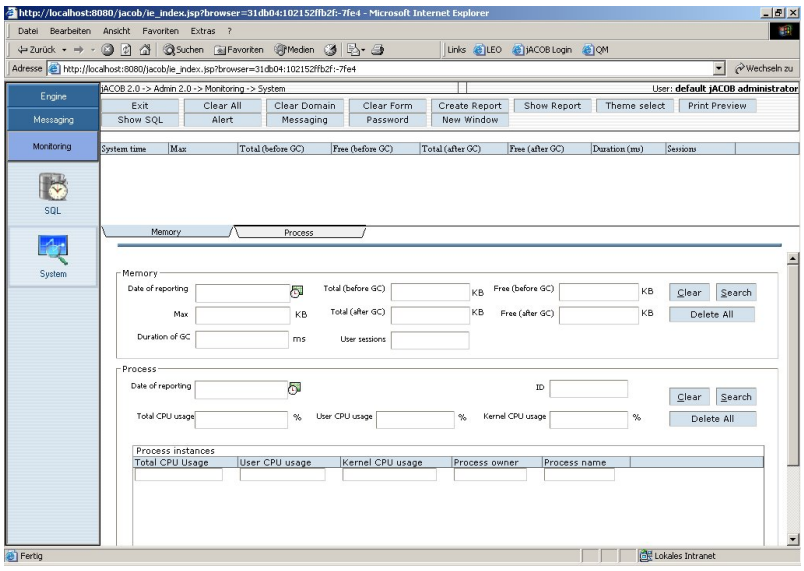


Figure 3.3. System form

The evaluation of the memory usage before and after the garbage collection (GC) is displayed in the Memory group:

- Date of reporting
- Memory Total (before GC) in KB
- Memory Free (before GC) in KB

	• Max heap size defined for the Java Virtual Machine • Memory Total (after GC) in KB • Memory Free (after GC) in KB • Duration of GC in milliseconds • The number of User sessions that were active at the end of the interval The snapshot of the processor load is displayed in the Processor group: • Date of reporting • Internal ID of the snapshot • Total CPU usage (kernel and user usage) in percent • User CPU usage in percent • Kernel CPU usage in percent • The table Process instances lists all processes that were mainly consuming the CPU at that moment in time
Emptying of the system monitoring database	The jACOB™ administrators are solely responsible for the regular emptying of the system monitoring database. This can be done by pressing the Delete All Records buttons.
Procedures	The following procedures will be described in the next sections: • Viewing the memory requirements

Viewing the memory requirements

Purpose

The observation of the memory requirements is a useful instrument to optimize the distribution of the system memory. The memory requirements in the past can be viewed with the aid of an unconstrained search in the Memory group. The search result can be exported to Excel and displayed as a meaningful chart.

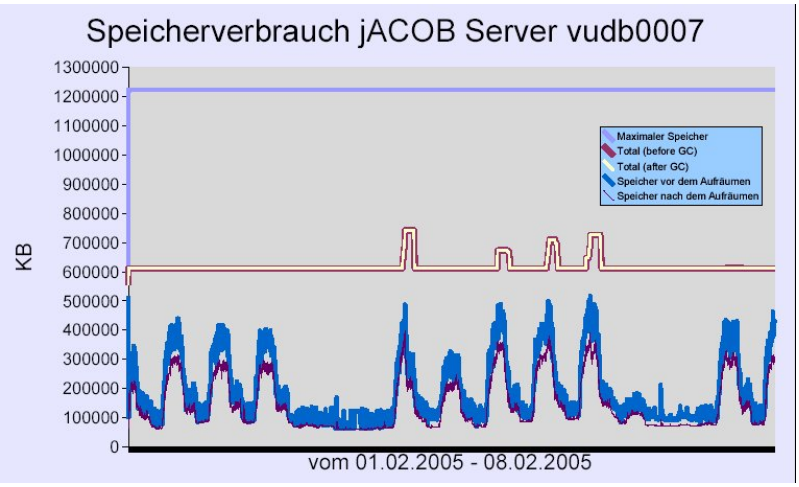


Figure 3.4. Excel chart of the memory requirements in the past

Impact

If a jACOB™ administrator observes the memory requirements, he/she can easily determine the memory requirements actually needed. Thus he/she can adapt the Max heap size defined for the Java Virtual Machine to its needs.

Procedure

Perform the following steps to view the memory requirements in the past:

Step	Procedure
1	Open the System form of the Monitoring domain.
2	Enter your search criteria into the form. E.g. a value of ">now-28d" in the Date of reporting field will search for records of the last four weeks. Press the Search button in the Memory group. All existing records matching the search criteria will be listed in the Memory search browser.
3	Export the data to Excel by clicking the Export to Excel  icon in the upper right corner of the browser. Format the data in Excel as desired.

Glossary of terms

B

backfill	Activity triggered by the selection of a data record in a search browser. The record data and the data of related records is entered, i.e. "backfilled", into the groups of the forms relevant to the record.

C

connection pool

A cache of database connections maintained in the memory of the application server so that the connections can be reused when the database receives future requests for data. Connection pools are used to enhance the performance of database requests. I.e. the overhead of building up and tearing down a connection for each request is omitted.

G

garbage collector
: A process periodically freeing the memory used by objects that are no longer referenced.

H

hot deployment	Hot deployment is the process of adding new components to a running server without having to stop the server process and restart it.

J

JNDI	Java naming and directory interface
JNDI data source	Data source located via JNDI.
jACOB application property	Configuration parameters specified for particular applications are stored in jACOB application properties.
jACOB application task	*jACOB task* defined by the application programmer.
jACOB internal task	Predefined *jACOB task* e.g. needed to maintain *user sessions*.
jACOB™ maintained data source	Data source that is maintained by the jACOB™ application server.
jACOB property	Configuration parameters specified for the application server in general are stored in jACOB properties.
jACOB task	Background job executed periodically within the jACOB™ application server.
jAN	jACOB™ Application Notifier, a message router for alerts, faxes, SMS messages and emails.

O

optimistic locking

Locking technique for data records. Also known as write locking. Allows unlimited read access to records. A record can only be written to the data source if the record has not changed since it was last read.

P

pessimistic locking	Locking technique for data records. Records are locked before they are edited, which ensures that only one user is editing the record at any given time.

S

stored procedure

In a database management system, a stored procedure is a set of SQL statements with an assigned name that is stored in the database in compiled form so that it can be shared by a number of programs.

U

user runtime property	Configuration parameters specified for specific users of specific applications are stored in user runtime properties.
user session	The activity session that a user with a unique IP address spends on a Web site during a specified period of time.