INSTITUTIONEN FÖR DATA-OCH SYSTEMVETENSKAP SU / KTH

DATABASES & INTERNET LABORATION

v. 1.1

*63

RELATIONSDATABASHANTERINGSSYSTEM

HÖSTTERMINEN 1999

http://L238.dsv.su.se/courses/stjarna63/



nikos dimitrakas

Sec.

Institutionen för Data-	Databases & Internet Laboration v. 1.1	Stockholm
och Systemvetenskap	*63	November 1999
SU/KTH	Relationsdatabashanteringssystem	
nikos dimitrakas		

Table of contents

.5 .8 .8
. 8 . 8
. 8
. 8
10
11
12
13
14
17
17 23
 17 23
 17 23 24
 17 23 23 24 27
 17 23 24 27 31
 17 23 24 27 31 32
 17 23 24 27 31 32 34
 17 23 24 27 31 32 34 35
 17 23 24 27 31 32 34 35 36
 17 23 24 27 31 32 34 35 36 36
1 1 1 1

Table of figures

Figure 1 Environment components and their role	4
Figure 2 Inside Websphere AppServer – Execution of a servlet	18
Figure 3 Inside ASP plug-in - ASP execution	32

1 Introduction to the Environment

In this exercise (described in chapter 2 & 3) we will build a little web application that provides the possibility to on-line retrieve and manipulate data stored in a database. We will provide the user with some simple database operations:

- Basic DML commands such as:
 - SELECT
 - INSERT
 - UPDATE
 - DELETE

We will try to do all this by using two different techniques:

- ASP (Active Server Page)
- Servlets and JSP (Java Server Page)

To do all this we will use the following products (they are all running under Windows NT, equivalent software may exist for other operating systems):

Netscape Enterprise Server 2.6	This is a web server product that provides the possibility to publish the applications on the web.
Chili!Soft ASP 3.0	This is a plug-in that activates the web server for ASP. Without this plug-in the ASPs are just text files for the web server.
IBM WebSphere Application Server 2.03	This is the application server that takes care of servlets, compiles jsp-files, creates and manages sessions and database connections. WebSphere Application Server is connected to the Netscape web server.
IBM WebSphere Studio 1.0	This tool provides help for designing servlets, javabeans, queries, and connections between all these. WebSphere Studio generates some of the code which can be completed in an editor of your choice. NetObjects Scriptbuilder is the default editor that is installed with WebSphere Studio.
Microsoft Access 97	This is the database manager that has been used to create the database used in the exercises in the following chapters.
ODBC Data Source Administrator	This is the "bridge" between the database and the ASPs and Servlets. ODBC Data Source Administrator can be found in the Control Panel.

The following figure illustrates how the web server, the application server, the ASP plug-in, and the database are connected to each other and to the internet. The web server is in charge of receiving requests from clients and then forwarding them to the application server, which is in charge of servlets and JSPs, or to the ASP plug-in which is in charge of ASPs.



Figure 1 Environment components and their role

1.1 Netscape Server Enterprise

On every machine (that is used for this course) there is a Netscape web server installed (normally under c:\Netscape\SuiteSpot). This means that the machine listens on port 80 for http requests. You can test this by starting a browser and trying the following URL: http://localhost/ or http://%nameofmachine%¹ for example http://L269.dsv.su.se/

If the browser returns an error message then the web server may be stopped. If the server is running the browser should show the default homepage:



¹ The name of the machine is usually "L" and a three digit number and it is written on the front part of the computer tower.

From here you can start and stop the web server.



An other very useful feature here is the possibility to create aliases for directories or files so that they can be accessed through http. For example we can create an alias (URL prefix) called star and map it to the d:\star directory:

Institutionen för Dataoch Systemvetenskap SU/KTH

nikos dimitrakas

💥 Netscape Enterprise S	Server - Netscape 📃 🗖	×				
<u>File Edit View Go</u>	<u>C</u> ommunicator <u>H</u> elp					
🔮 💕 🌏 Back Forward Reload	A A A A A A A A A A A A A A A A A A A	\mathbb{N}_{-}				
🚪 🦋 Bookmarks 🮄 Netsite	e: http://l269.dsv.su.se:14614/https-l269/bin/index	ed				
📱 🖳 SMS-MTN 📹 Channels-	-Tävl 📫 Search Engines 📫 Myndigheter 📫 Multimedia.,mm 📫 Diverse 📫 Music 📫 Help in sublect 📫 Diverse help 🖆	ј Вог				
NETSCAPE [®] EN © 1999 Netcope Con Server Programs	TERPRIST SERVES 3.6 Apply t. Admin multicating Configure on M. Content of Web on Agents & Autop					
	Ovyles management in bishing Search Catalog					
Content	Additional Document Directories					
Management	Add Another Directory					
Primary Document						
Directory						
Additional	URL prefix:					
<u>Document</u>	http://1269.dsv.su.se/ star					
<u>Directories</u>	New The Disastern division					
Remute File	Map To Directory: d:/star					
<u>Manipulation</u>	Apply Style: NONE					
<u>Document</u>						
Preferences	OK Reset Help					
URL Forwarding						
Hardware Virtual						
<u>Servers</u>	Current & differed Directories	1				
Software Virtual						
<u>Servers</u>						
International	Edit Remove Prefix: /caspagent					
<u>Characters</u>	Directory: C:/Program Files/ChiliASP/Samples/friendship/agent/content					
Document Footer	Edit Remove Prefix: /caspclient					
Parse HTML	Directory: C:/Program Files/ChiliASP/Samples/triendship/client/content					
Cache Control	Edit Remove Prefix: /casp401k					
Directives 🔽	Directory: C:/Program Files/ChiliASP/Samples/401k/content					
Docume Docume	ent: Done 🔤 🗰 😼 💷 😰 🖬 🎸					

After pressing OK and applying the changes it will be possible to see the contents of the d:\star directory with a web browser at the following URL:

http://L269.dsv.su.se/star/

赛In	ndex of	/star/ ·	- Netsc	аре								_ 🗆 🗙
<u> </u>	e <u>E</u> dit	<u>V</u> iew	<u>G</u> o	<u>C</u> ommu	unicator	<u>H</u> elp						
	🔮 Back F	orward	3 Reload	🚮 Home	a Search	Metscape	d Print	💕 Security	Stop			N
1	🛃 Bookma	ırks 🤳	Netsite:	http://L269.	.dsv.su.se/st	ar/				- 🕻) [*] What's I	Related
1	SMS-MT	'N 🖆 Ch	annels - T	ävl 📹 Se	earch Engine	s 🖆 Mync	ligheter 🖆	Multimedia	a,mm 付	Diverse	🖆 Music	: 🖆 Help
Ir	ndex	of	/sta	r /								
	Name			I	ast mod	ified	Size	Descr	iption			
	Parent file. test.t	t Dired html txt	ctory	1	8-Nov-99	9 11:27 9 11:26	0ĸ 0ĸ					
d – 1	Þ	D	ocumer	nt: Done						4.0 e	19 🖬	🥩 //.

1.2 WebSphere Application Server

To every Netscape Server there is a WebSphere Application Server attached. The WebSphere Appserver is located under e:\WebSphere\AppServer. The servlets that we will create later have to be deployed on the WebSphere application server. WebSphere searches automatically for .class (compiled java classes) and .servlet (servlet configuration) files in the e:\WebSphere\Appserver\classes directory and the e:\WebSphere\Appserver\servlet directory.

WebSphere Application Server can be administered at port 9527: <u>http://localhost:9527</u>

username: *admin* password: *admin*

Useful information about WebSphere Application Server is also available at: <u>http://localhost:9527/doc/index.html</u>

1.3 Chili!Soft ASP

Chili!Soft ASP is a plug-in that is also attached to the web server. This product does not require any configuration.

For more information about Chili!Soft ASP: <u>http://localhost/caspdoc/Index.html</u>

1.4 WebSphere Studio

WebSphere Studio is a tool for creating most of the necessary files for web applications based on servlets and JSPs. Studio can also be configured so that it can "publish" all the publishable files (what it basically does, is copy them) to predefined directories where WebSphere Application Server and Netscape Server can find them. That can be set in the Options > Publishing servers menu:



Publishing Server Settings	🔣 Edit Publishi	ng Server Setting	js 🗙
Select a publishing server and click Edit, Delete, or Duplicate. Or, click Add to define settings for a new publishing server.	Name Local C Remo Host Settings Host Loserid Password	te © Local	
	Publishing targets —		
OK Cancel	Name	Path	Add
	html classes	c:\WebSphere\Studi c:\WebSphere\Studi	Remove
	jsp	C:\WebSphere\Stud	Browse
		Ŀ	
	Advanced	OK Cancel	Help

Here you can define the "publish"-paths for different types of files².

All classes and servlet files should be published to c:\Websphere\Appserver\classes

All html, jsp, gif, jpg etc. should be published either to c:\Netscape\Suitespot\docs (*which is the default document root for the web server*) or to a directory that you have mapped with an alias from the web server (see section 1.1 on page 5).

WebSphere works with projects. Every project is saved in a directory with the name of the project under c:\Websphere\Studio\projects. This location can be redirected:

WebSphere Studio	
<u>File Edit View Options Tools H</u> elp	
→ B I I Preferences to to	Preferences
File extension	rectories Compiler Help
Classes <u>Welcome</u> ↓ class ↓ class ↓ tar63	Project directory c::///ebSphere/Studio/projects Browse
book F	roject template directory c://webSphere\Studiottemplates\project Browse Browse Browse Browse
🖉 🖉 owne	
🖉 🍕 autho	OK Cancel Help
6 objects 4233 bytes	

 $^{^2}$ Notice that a whole folder is assigned to a publishing location and that all the publishable files are published to the associated location. You don't need to worry so much about the publishing, Websphere Studio is quite good at publishing the right files to the right place.

Together with the directory where the project files are located, there is two more configuration files:



You can back-up your project by copying these three items (project directory & project configuration files).

WebSphere Studio also provides Wizards that guide you through the most common steps of creating JavaBeans, servlets, SQL statements, html and jsp files. We will explore these features in more detail later.

The documentation of WebSphere Studio can be accessed with a web browser at: c:/WebSphere/Studio/DOC/html/2tabcontents.html

1.5 Other tools

Other relevant tools are:

- ScriptBuilder for editing .html, .jsp, .java, .servlet, .asp files ScriptBuilder provides some support on those types of files.
- MS Access to explore and alter the database The database used in the following exercises can be downloaded from
 - http://L238.dsv.su.se/courses/stjarna63 or
 - \\DB-SRV-1\StudKursInfo\x63 Ht1999\ASP-Servlet Laboration

ODBC Data Source Administrator

For creating an ODBC alias for a database so that servlets and ASPs can access the database. The ODBC Data Source Administrator can be opened from the control panel. When you register an alias make sure to place it under the <u>System DSN</u> tab, so that the servlets and ASPs can find it:

ystem Da	a Source	Add
Name Flowers	Microsoft Access Driver (*.mdb)	<u>B</u> emove
FlowerSho Pohool	p Microsoft Aucess Driver (*.mdb)	Configuro
	An ODBC System data source stores informa	ation about how to connect to the

2 Database

In chapter 3 and 4 we will build two small applications that connect to a database with information about books, authors and book owners. This database is built in MS Access and includes some test data. The following figure illustrates the tables of the database as well as the relationships between them:



Institutionen för Data och Systemvetenska SU/KTH nikos dimitrakas	a- Databases & Internet Laboration v. 1.1 ap *63 No Relationsdatabashanteringssystem	Stockholm ovember 1999
Owner	Includes data about people that own books	
Publisher	Includes data about publishers	
Author	Includes data about authors	
Book	Includes data about books. A book is identified by its na author.	ame and main
AuthorsOfBook	Includes data about additional authors of books. This ta added in order to remove the many-to-many relationship b and authors.	able has been between books
BookItem	Here we have all the copies of a certain book with their a differ from copy to copy) for example their owner.	attributes (that

2.1 Connect a database

To use the database in our web applications we need to do the following things:

- 1. Download the database from
 - http://L238.dsv.su.se/courses/stjarna63 or
 - \\DB-SRV-1\StudKursInfo\x63 Ht1999\ASP-Servlet Laboration
- 2. Create an ODBC alias (also known as DSN) in the ODBC Data Source Administrator.



ODBC Microsoft	t Access 97 S	etup			×	
Data Source <u>N</u> ame	lab63			ОК		
Description:	database used	for the lab of *63		Canco	. 1	
Database					·	
Database:		Select Databas	e	Heln		X
<u>S</u> elect	<u>C</u> reate	Database Name		Directories:		
		book.mdb		d:\my documents\	stiëma 63	
- System Database -		backup.book.mdb	<u> </u>	🗁 d:\		Cancel
⊙ Non <u>e</u>		book.mdb		My documen	ts	<u>H</u> elp
O Database:			Select the that you h	database file ave		□ <u>R</u> ead Only
	System Databi		download	ed.		<u> </u>
					Y	
		Access Databases	(unuu -)	u seconara	ution 💌	Network
		•				

Now the database is available through an ODBC driver.

3 Servlets

In this section we will build a project with servlets and JSPs. The following two functions will be available:

- List all the books (title, mainauthor, isbn) and their publisher (name and country) ordered by publisher, author and book title.
- Update the information of an author.

To do that we will use WebSphere Studio and its Wizards.

We can start by creating a new project in WebSphere Studio.



The name of the project used throughout this compendium is *test63*. We will use this project as a container for all the servlets and JSPs that we will create. When the new project is created, WebSphere Studio automatically creates a directory for the entire project and in that directory another directory for all the classes.

🔛 WebSphere Studio			×
Eile Edit Yiew Options Tools Image: Second se	Help	Project location	
	c:WVebSphere\Studio\projects\test63		_
E@ test63	Nerrie	Size Modified	Ŷ
🛄 classes	💐 classes	0	
	Class	directory	
		4	
	<u> ا</u>	•	T.
1 objects 0 bytes			

3.1 Using the SQL Wizard

When you want to create a servlet that provides database functionality, you first have to specify an SQL statement and some database configuration parameters (user-name, password, driver...). The SQL Wizard provides that.

 ✓ Create a servlet that returns a table with the results of the following request: List all the books (title, mainauthor, isbn) and their publisher (name and country) ordered by publisher, author and book title.

To do that we can use the SQL Wizard to create an SQL statement that complies with that request.

- Start the SQL Wizard! (*Tools* > *SQL Wizard* or *the button on the right top side*)
- Name your query! (For example select1)



> Press Next and fill out the form according to the next figure:

SQL Wizard - c:\WebSph Welcome Logon Finish	ere\Studio\projects\	\test63\select1.sql
	Connect to a database Database URL: [jdbc:odbc:lab63] Userid: Driver: JDBC-ODBC Bridge	Password: Other driver: Sun.jdbc.odbc.JdbcOdbcDriver
Enter the required information and cli	ck 'Connect' to begin.	
		< Back Next > Finish Cancel

Press the Connect button to get to the next step. If the connection is successful then a list of all the available tables should come up:

🜺 SQL Wizard - c:\WebSphere\Studio\projects\test63\se	lect1.sql	_ 🗆 ×
Welcome Logon Tables Join Columns Condition 1 Sort SQL Finish		
Select an SQL statement type and table(s).		
Ctotomont type:		
💽 Select 🔘 Select Unique 🌔 Insert 🌔 Update 🌔 Delete		
Select Table(s):		
Table name		
MSysACEs		
MSysModules		
MSysModules2		
MSysObjects		
MSysQueries		
MSysRelationships		
	Filter	table(s)
,		
< Back	Next > Finis	h Cancel

- Select the tables that should be involved. (in this case book and publisher)
- Press Next to go to the "Join" tab. Here you can define rules for joining the two tables (basically for the foreign keys).

Show how the tables are related by joining them.	SQL Wizard - c:\WebSphere\Studio\projects\test63\select1.sql
Joined 'book.publisher' and 'publisher.pname'.	Show how the tables are related by joining them.

- Press Next to go to the "Columns" tab.
- Include the columns that are requested. (e.g. book.title, book.mainauthor, book.publisher, publisher.country, book.isbn)

SQL Wizard - c:\WebSpher Welcome Logon Tables Join Columns Select the columns you wish to include.	e \Studio \pro	jects \test63 \select1.sql _ C	
Selected table(s): publisher Columns: pname address postalcode city phone Intervention 	Add >> << Remove Select all Deselect all	Columns to include: book.title book.mainauthor book.publisher book.isbn publisher.country Move up Move down	-
		<back next=""> Finish Ca</back>	ncel

- > Press Next to go to the "Condition" tab.
- You can leave the conditions empty.

- ➢ Press Next to go to the "Sort" tab
- ➤ Here you can define which column(s) the result should be ordered by.

Icome Logon Tables Join Select how you want the resulting	Columns Condition 1 Sort	SQL Finish
Columns:		Sort order:
book.isbn		Ascending
publisher.country		Columns to sort on:
		(a->z) book.publisher
	Add >>	(a->z) book.mainauthor
	are Domous	(a-≻z) book.title
	<< Remove	
	Select all	
	Deselect all	
	Doooloot dii	Move un Move down
		more down
,		

> In the next tab you can see the SQL statement that you have graphically built.

➢ You can now hit the "Finish" button.

A new file has been added to the project containing all the information about the SQL statement:

🙀 WebSphere Studio	👷 WebSphere Studio 📃 🗆 🗙											
<u>File Edit View Options Tools</u>	<u>H</u> elp											
	c:\WebSphe	re\Studio\projects\test63										
⊡∽🧶 test63		Name	Size	Modified	Modi 🔶							
🔲 🥘 classes	<u></u>	classes	0									
	Ø 🕹	select1.sql	1409	1999-nov-23 15:32:33								
					桑							
					۹							
					۹							
	•				•							
1 objects selected 1409 bytes	·											

3.2 Using the Studio Wizard

Based on the SQL created by the SQL Wizard, the Studio Wizard can create a servlet, a JavaBean and JSP. The following figure illustrates how these three components work together to provide the web client with the result.



Figure 2 Inside WebSphere Application Server – Execution of a servlet

- Start the Studio Wizard. (Tools > Studio Wizard)
- > Choose to create a Database access servlet and press Next.



Give the servlet a name. (e.g. selectservlet)

🔛 Database Access Wizard 📃 🗆 🗙						
Welcome SQL statement Input page Input layout Output page Output layout Custom error page Finish						
	This wizard will create a servlet that accesses database information. To get started, name your servlet and click Next. Name Selectservlet Java package test63 Comment first servlet. uses a simple select statement Output folder c:\WebSphere\Studio\projects\test63 Browse					
	<back next=""> Finish Cancel</back>					

- (1) The Java package property is normally the same as the project name. In projects with many servlets it can be good to have a better structure of java packages.
- ➢ Press Next.
- > Choose the SQL statement that was created by the SQL Wizard before.

👪 Database Access Wizard	_ _ ×
Welcome SQL statement Input page	Input layout Output page Output layout Custom error page Finish
	Which SQL statement would you like to use? Select from an active project Look in all open projects Select from the file system Browse
	< Back Next> Finish Cancel

> Leave the next four tabs as they are and go to the "Finish" tab.

🔛 Database Access Wizard	1	
Welcome SQL statement Input page	Input layout Output p	age Output layout Custom error page Finish
	Your database access	s servlet is now complete.
	Click Finish to save yo	ur files and return to IBM WebSphere Studio.
	You can change the fo	llowing generated filenames:
	Servlet	SelectServlet.java
	Data access Bean	SelectServletBean.java
	Input page	SelectservletinputPage.html
	Output page	SelectservletOutputPage.jsp
		Back Next Finish Cancel

Here you have the possibility to choose the filenames for all the components that are about to be created. The Input page is just an html page with a button that calls the servlet. The Data access Bean is a javabean that makes the connection to the database and executes the SQL statement. The Output page is a JSP file that gets the information from the javabean and formats it into plain html. The servlet is the connection between all the other components. It receives and handles a request, instantiates the javabean and initiates a response (see Figure 2).

Press Finish!

All the files have now been created and the classes have been compiled. It is now possible to edit these files, for example to make design changes. For now we will not make any changes. The only thing remaining is to publish the files.

➤ Mark the project, right click on it and choose Publish...

Publish

Cancel

Help

V 🔛	VebS	phere	Studio)								_ 🗆 ×
File	<u>E</u> dit	⊻iew	<u>O</u> ptions	<u>T</u> ools	<u>H</u> elp							
Ð	ß	e é	D d	B O	6	⁺⊔	* * *				2	
					c:Web	Sphe	re\Studio\projects\test63					
•	🔰 test6:	Duki	iola				Name	Size	Modified	H Modifi	ed By	Owne 🔶
		Refr	isri esh			0	classes	0				۲
		Save	project		0	۹	select1.sql	1409	1999-nov-23 15:3	32:33		
		 <u>C</u> los	e project) 🧇	SelectservletOutputPage.jsp	1275	1999-nov-26 11:5	59:44		2
		Copy	(0		SelectservletInputPage.html	404	1999-nov-26 11:5	59:45		8
		Past	e •-				🔆 Select Publishing Serv	er	×			٢
		Dele	te ortioc	-			Select the publishing server to use a	and click Continue.				3
			entres				Or, click Edit to change the settings	for the selected serv	er			
							Servir 1238	Edit				
							Cill_zon Homonogolda					
							Total number of files to publish: 5	Details		X		
					•		Cont	inue Cancel	🔆 Publish ta	1238		
4	4 object	s 3088	bytes						Verify the nublicat	hing covor cottings		
									Change these va	alues to temporarily o	verride the sto	ored settings.
~	Cha		h Du	h1:	C	~	an that way have		Nama	Path		Eilee
	Cno	ose i	ne Pu	IDIISIII	ng S	erv	er that you have		html	C:\L238-Homepage\	docs	2
	conf	ïgur	ed (se	e secti	on 1	.4).			classes	C:\WebSphere\AppS	erver\classes	3
<i>b</i>	Pres	s Co	ntinue	e and f	hen]	Puh	lish					

You should get a message that you may need to restart your web server.

➢ Press OK.

It should now be possible to run the servlet.

 Start a web browser and go to the following URL: <u>http://%nameofmachine%/%InputPagePath%%InputPageFilename%</u>
 Where %nameofmachine% is your machine's name, e.g. L269, %InputPagePath% is the path to your input page, starting from the web server's root. In this case nothing, %InputPageFilename% is the name of the input file generated by WebSphere Studio. In this case SelectservletInputPage.html.

For example <u>http://l238/selectservletInputPage.html</u>:

嶶	Netsc	ape									_	
Ei	ile <u>E</u>	dit <u>V</u> iev	v <u>G</u> o	<u>C</u> ommu	inicator	<u>H</u> elp						
F mm	- 🌒	2	3		ø.	m)	3	i 🗳				N
	Back	Forward	Reload	Home	Search	Netscape	Print	Security	Stop		_	
<u>à</u> 1	🌿 🐻	okmarks 🙏	Netsite:	http://L238/	/selectservi	letinputpage.	html			- 🗊	"What's Re	elated
Ξ.	🖳 SMS	-MTN 🖆 C	hannels - T	Fävl 📹 Se	arch Engin	es 📹 Myno	digheter	📺 Multimedi	ia,mm 付	Diverse	📑 Music	He
Pl	ease c	omplete th	e form a	and click	Submit.							
	Submit	Reset										
	=0=		Documei	nt: Done						4 <u>8</u> d	ې 🔝 🦻	v //

➢ Press Submit.

Output page for servlet Bean: test63.SelectServlet - Netscape										
<u>File Edit View Go Commu</u>	nicator <u>H</u> elp									
Back Forward Reload Home	🤌 🚵 📑 Search Netscape Print	ecurity Stop		N						
🤯 Bookmarks 🎄 Netsite: http://L238/servlet/test63.SelectServlet 🔍 💎										
🚪 🖳 SMS-MTN 🖆 Channels-Tävl 🖆 Sea	rch Engines 🧉 Myndigheter 🛛	🖆 Multimedia,mm 📑 Diverse 📑	Music 🖆 Help in sublect 📫 D	liverse help 📫 BookStores [
title	mainauthor	publisher	isbn	country						
Concepts of programming languages	Robert W. Sebasta	Addison-Wesley	0-8053-7133-8	USA						
Sångfågeln från Milano	Annette Kullenberg	Bra böker	n/a	Sweden						
Rabbit Redux	John Updike	Forum	91-37-09391-6	Sweden						
Rabbit Run	John Updike	Forum	91-37-09390-8	Sweden						
Management Systems Conceptual Considerarions	Peter P. Schoderbek	Irwin	0-256-25166-5	US						
Dödliga Data Håkan Borgström		Natur och kultur	91-27-02803-8	Sweden						
System Analysis and Design	Kenneth E Kendall	Prentice-Hall International	0-13-654229-8	UK						
Vägen till C	Ulf Bilting	Studentlitteratur	91-44-26732-0	Sweden						
Document: Done				🔆 🛂 🗗 🖬 🏑 🎢						

This is the default layout of the output page. It can easily be changed by editing the selectservletoutputpage.jsp file. To edit it you can double click on the filename in WebSphere Studio.

A jsp file is very similar to an html file. In addition to the usual html tags there are tags that help to access the javabean. Such tags appear in selectservletoutputpage.jsp:

<BEAN> <REPEAT> <INSERT> <% %>

When editing jsp files be careful with those tags. If they are not correct, the server will return an error. After making changes you have to republish the altered file(s).

3.3 Updating data

Updating data can be a little more complicated. WebSphere Studio cannot create all the components needed to insert, update and delete data.

In this section we will create three servlets for updating the information of an author:

- 1. One that shows a list of all the authors in the database so that the user can choose one.
- 2. A second servlet that shows the information stored in the database for the selected author, so that the user can change it.
- 3. A final servlet that commits the changes to the database.

The first two can more or less be generated by WebSphere Studio. The third servlet requires a little coding.

3.3.1 Showing all authors

This part is similar to sections 3.1 & 3.2.

- Create in a similar way a servlet that shows the names of all the authors.
 - ① Create an sql with the SQL Wizard.
 - ① Create a database access servlet with the Studio Wizard.
 - ① Use smart filenames, think that there is going to be around 15 files.
 - Don't forget to publish your project.

When you are done you should have a servlet that produces a result similar to the following:

💥 Output page f	for servlet	Bean: te	est63.U	pdate1Se	rvlet -	Netscap	e	_ 🗆 ×
<u>Eile E</u> dit <u>V</u> ie	ew <u>G</u> o	<u>C</u> ommu	nicator	<u>H</u> elp				
i 🗳 🚽 🔊	3		_ <u>æ</u>	M	4	_ 🗳 _		N
Back Forward	I Reload	Home	Search	Netscape	Print	Security	Stop	
🛛 🦣 BOOKINAIKS 🤉	i Channels - Të	mp:///238/s ivl 🗂 Sea	arch Engin	es 📬 Mvndi	arviet iaheter i	× I ▼ Multimedi	a.mm 📬	Diverse (
			ş		5			
	aname							
		A	aron Sp	elling				
		Ann	iette Kul	lenberg				
		A	nnika B	omark				
		Arth	ur Cona	n Doyle				
		Aste	erios G.	Neraras				
		Chorl		hodorbolz				
		D	es G. St	doma				
			drfe	suams				
		Em	est Hem	inoway				
		Fr	nesto Sa	mtiaon				
			Junnel I	inde				
		He	enrik Ko	estner				
		Hå	kan Bor	gström				
		J	ames W	oods				
		Ja	an Skans	sholm				
			John Up	dike				
		Ju	lie E. K	endall				
		К	arsten A	Inaes				
		Ker	meth E l	Kendall				
		nil	cos dimi	trakas				
		Pete	r P. Sch	oderbek				
			Roald I	Dahl				
		Roł	pert W.	Sebasta				
			UlfBil	ting				
						_		
	Document	:: Done				🛞 🍇	1 S	1

3.3.2 Showing selected author's information

The second step is to show the information of the selected author. We can create an SQL with the SQL Wizard that uses a parameter:

- Create a new sql with the SQL Wizard in the same way as before until the "Condition" tab.
- > At the "Condition" tab, select the column that is going to be in the condition (the aname).
- Select the condition operator (equal to).
- > Press the parameter button and give the parameter a name.
- ➢ Press OK.

SQL Wizard - c: Welcome Logon Tables	\WebSphere\Studio\projects\i Join Columns Condition 1 Sort SQL Finis	test63\update2.sql	
Select a column, an op	erator, and enter the values you want to find.		
Selected table(s):	Distrator.	Values:	
author	is exactly equal to	_	
AUUIMINS	is after		
aname	is before		
balletere country	is before or equal to is between		
Country	contain the character(s)		
	end with the character(s)		
	is blank is not blank	Find Parameter Clear	
In table 'author',		Eind on enother blymn	
find all rows in column) 'aname' that		
are exactly equal to		V	🙀 Create a new par 🗵
			Enter the parameter name below:
,			
		< Back Next> Finish Cancel	OK Cancel Reset

- Continue with the SQL Wizard as before.
- ➤ Create a database access servlet as before.

Now it is time to write some code!

WebSphere Studio has generated a javabean that has code for activating the parameter in the sql. That code works with most databases but not with MS Access. Therefore that code has to be replaced with some more primitive embedded SQL.

> Open the javabean in an editor (by double-clicking on it)

Here there is the following code:

```
/**
 * Instance variable for the SQLString property
 */
protected java.lang.String SQLString = "SELECT \"author\".\"aname\", \"author\".\"birthdate\",
\"author\".\"country\" FROM \"author\" WHERE ( ( \"author\".\"?\" = aname ) )";
...
// Create placeholders for the parameters
metaData.addParameter("aname", java.lang.String.class , 12);
...
initialize();
```

// Initialize the parameters for the query
selectStatement.setParameterFromString("aname", getAname());

The SQLString variable contains the SQL statement to be executed. In that statement there is a question mark. That question mark is supposed to be replaced with the value of the aname variable (the parameter). This technique is not accepted by MS Access.

Instead we can build the SQLString first and then execute it:

- Scroll down to the method called initialize().
- Remove the following rows of code:
 // Create placeholders for the parameters metaData.addParameter("aname", java.lang.String.class, 12);
- Scroll down to the method called execute().
- Remove the following rows of code:
 // Initialize the parameters for the query selectStatement.setParameterFromString("aname", getAname());

To replace the rows that we just removed we can add the following code to the method execute(), exactly before this row

initialize();

Add the following:

SQLString = "SELECT author.aname, author.birthdate, author.country FROM author WHERE author.aname = '" + aname + "'";

In this way we have built the SQLString before we initialize the database connection and MS Access cannot complain.

- Save the file and compile it in WebSphere Studio (Mark the file and choose File > Compile file).
- Publish the project (If you have already tried to run the servlet before the changes, then you may need to restart the web server (see section 1.1)).

After all that is done, you should have a servlet with the following input and output pages:

	🔆 Out	put page fo	r servle	t Bean:	test63.U	pdate2Se	rvlet -	Netscape	5	_ 🗆 ×
🕱 Netscape 📃 🗆 🗙	<u>F</u> ile	<u>E</u> dit <u>V</u> iev	v <u>G</u> o	<u>C</u> omm	nunicator	<u>H</u> elp				
<u>File Edit View Go Communicator H</u> elp	1	· 🏹	3	1	ø.	my.	4	aî.	21	N
🧉 🔉 🤉 🔥 🙇 🛍 N	Ba	k Forward	Reload	Home	Search	Netscape	Print	Security		
Back Forward Reload Home Search Netscape 📠	1	Bookmarks 🤳	Netsite:	http://1238	3/servlet/test	t63.Update2S	ervlet		▼ () * Wi	nat's Related
👔 🍑 Bookmarks 🙏 Netsite: http://l238/updat 🖵 🕼 What's Related	_ 🖪 S	/IS-MTN 📺 (hannels - T	Fävl 🖆 S	Search Engin	es 📫 Mynd	igheter (ゴ Multimedia	a,mm 🖆 Diverse	e 📫 Music 🕻
📱 🖳 SMS - MTN 📫 Channels - Tävl 📫 Search Engines 📫 Myndigheter		anam	0		hi	rthdato			country	
Disease complete the form and alight Submit		D 1 4	1		1052.01	01.00.00	00.0		T 1 1	
Prease comprete me form and crick Subinit.		Douglas A	dams		1952-01-	01 00:00:0	00.0		England	
ana ne Douglas Adams										
			•							
Submit Docot										
Submit Reset										
🖻 🗝 🔰 Document: Done 📃 💥 😼 🚳 💋 🎼			Documei	nt: Done					🧩 🍇 🐠	🖼 🥩 //.

Now we have to connect these to servlets. The user should use the output page of the first servlet as an input page to the second. Similarly the output page of the second servlet should

Institutionen för Data-	Databases & Internet Laboration v. 1.1	Stockholm
och Systemvetenskap	*63	November 1999
SU/KTH	Relationsdatabashanteringssystem	
nikos dimitrakas		

be used as the input page to the third servlet. To do this we need to "merge" the output page of the first servlet with the input page of the second servlet.

The input page of the second servlet is quite simple. It is an html file with a form. The form's action calls the second servlet. The form has an input field for the parameter and a submit button.

The output page of the first servlet has a loop (<REPEAT>) for going through the result of the SQL.

By combining these two we can get an output page that looks like this:



3.3.3 Updating data

The third servlet is a little different. Instead of a SELECT statement it should contain an UPDATE statement. Even though WebSphere Studio cannot create all the code needed, it is a good idea to use the Wizards to create as much as possible. To do that we can create an SQL statement (SELECT statement) that takes as parameters the same parameters as the UPDATE statement would take (oldaname, newaname, newbirthdate, newcountry - ID for the old record and all the new information).

- Create an SQL with the SQL Wizard.
- Set multiple condition by clicking on the "Find on another column" button
 - ① Since the SQL Wizard does not allow you to create conditions on date columns, set the condition on another column. We would anyway change the code... It doesn't matter if the sql doesn't make any sense right now. We will just use this sql statement to create a servlet and a javabean with parameter variables.
- > Create now a database access servlet with the Studio Wizard based on the new SQL.

We need to make the following changes to the generated files:

- 1. Change the javabean. Instead of a SELECT statement we should have an UPDATE statement. There is a lot of code that can be removed: all the methods that have to do with the result set (an UPDATE statement does not produce a result set).
- 2. Once there is no result set, there is nothing to show on the output page. We can modify that page to some static html.
- 3. Merge the input page with the output page of the previous servlet.
- Start by opening the javabean for editing.

All the following variables and methods have to do with the result set and are therefore of no use. Remove them:

private static final int author_aname_COLUMN private static final int author_birthdate_COLUMN private static final int author_country_COLUMN protected java.lang String SQLString public java.lang.String getSQLString() public java.lang.Object getAuthor_aname(int row) public java.lang.Object getAuthor_bithdate(int row) public java.lang.Object getAuthor_country(int row) public java.lang.Object getAuthor_country(int row) public java.lang.Object valueAtColumnRow(int column, int row) public void closeResultSet() protected conf.lbm.servlet.connmgr.IBMJdbcConn getPooledConnection(java.lang.String driver, java.lang.String URL, java.lang.String userID, java.lang.String password)

protected com.ibm.servlet.connmgr.IBMJdbcConn getPooledConnection(java.lang.String poolname, java.lang.String driver, java.lang.String URL, java.lang.String userID, java.lang.String password)

There are also a few variables that take care of the database connection:

protected com.ibm.servlet.connmgr_IBMConnMgr connectionManager; protected com.ibm.servlet.connmgr.IBMJdbcConn ibmJdbcConn; protected com.ibm.db.selectStatement selectStatement; protected com.ibm.db.SelectResult result;

We will replace them with variables from the java.sql package:

static protected Connection con; static protected Statement stmt;

Add the following line at the top of the file:

import java.sql.*;

Also remove the following method:

public void initialize()

This method established the connection based on the ibm classes. We will now add a method for establishing a connection based on the java.sql classes:

```
Institutionen för Data-
                              Databases & Internet Laboration v. 1.1
                                                                                     Stockholm
och Systemvetenskap
                                                *63
                                                                               November 1999
SU/KTH
                               Relationsdatabashanteringssystem
nikos dimitrakas
// establish DB connection
 public void dbConnect()
 {
   try
   {
    // register the driver with DriverManager
    Class.forName(getDriver());
    con = DriverManager.getConnection(getURL(), getUserID(), password);
    con.setAutoCommit(true);
   }
   catch (Exception e)
   {
     e.printStackTrace();
   }
 }
```

This method simply creates a database connection.

The main method of this javabean is the execute() method. It is this method that the servlet calls to execute the SQL statement. We can change this method to first call the dbConnect() method and then execute the UPDATE statement.

```
\triangleright Remove all the code of the method:
   initialize():
   // Initialize the parameters for the query
   selectStatement.setParameterFromString( oldaname",);
   selectStatement.setParameterFromString("newaname",);
   selectStatement.setParameterFromString("newcountry",);
   selectStatement.setParamerFromString("newbirthdate",);
   // Execute the SQL statement
   selectStatement_execute();
   result = selectStatement.getResult();
   // release the connection for use by another SQL statement
   ibrudbcConn.releaseIBMConnection();
➤ And replace it with this:
   dbConnect();
   String query;
   query = "UPDATE author SET aname = '" + getNewaname() + "', country = '" +
       getNewcountry() + "', birthdate = CDATE(LEFT('" + getNewbirthdate() + "', 10)) WHERE
       aname = '" + getOldaname() + "'";
   stmt = con.createStatement();
   stmt.executeUpdate (query);
   stmt.close();
   con.commit();
```

- ① CDATE and LEFT are functions that MS Access has. CDATE transforms a string to a date. LEFT returns a sub-string. They can be omitted if there is some other code that controls the length of the date parameter. Dates in MS Access are not exactly compatible with the java.sql.date format.
- Also change the declaration of the method to throw the appropriate exception: public void execute() throws java.sql.SQLException

When all this is done you should be able to compile the javabean without errors. If you would now try to execute the servlet, it would return the output page which would complain. That is because the output page is still configured according to the old javabean.

The last thing is to connect the second servlet to the third.

Here is an example of how to merge the output page of the second servlet with the input page to the third servlet:

			oldaname	
aname	birthdate	country	newaname	
Arthur Conan Doyle	1859-05-22 00:00:00.0	Scottland	iiv (cuidiliv	
			newcountry	
			newbirthdate	
			Saomit Reset	
		\uparrow		
	Please co	omplete the form	and click Submit.	
	Name: A	rthur Conan D	oyle	
	Birthdate	: 1859-05-22	00:00:00.	
	Country:	Scottland		
	Submi	t Query		

Please complete the form and click Submit.

① The field "oldaname" is also present in the new form but as a hidden field!

After publishing the project you should have a chain of these three servlets:

 Output page for servlet Bean: test63.Update1Servlet ... _ _ X

 File
 Edit
 Yew
 Go
 Communicator
 Help
 _ 🗆 🗙 <u>G</u>o <u>Communicator</u> <u>H</u>elp A Home Metscape Set N 🔌 Back A Home Metscape I N Set 3 Reload 📡 🧟 Forward Reload <u>ک</u> I d 🗳 Print 🯄 Search a Search I Print Back 🎸 Bookmarks 🙏 Netsite: /update1servletinputpage.html 👻 🕼 What's Related 🐠 Bookmarks 🙏 Netsite: servlet/test63.Update1Servlet 🔻 🕼 What's Related 🖳 SMS - MTN 🧉 Channels - Tävl 🖆 Search Engines 🖆 Myndigheter 🖆 Multimed 🖳 SMS - MTN 🖆 Channels - Tävl 🖆 Search Engines 🖆 Myndigheter 🖆 Multimedi Please complete the form and click Submit. Please complete the form and click Submit. Submit Reset Author name: 🛛 ▼ Submit ₫ ⇒⊫ Document: Done 🔆 😼 🖉 🖬 Document: Done 🍇 🕼 🖬 Ý Ż - 💥 ∰Output page for servlet Bean: test63.Update2Servlet ... _□× ∰Ou File Edit View Go Communicator Helo put page e for servlet Bean: test63.Update2Servlet ... 💶 💌 <u>File Edit View Go</u> Communicator <u>H</u>elp Edit View Go Communicator <u>H</u>elp Sec 3 Reload Form ð 3 **☆** Home My. m) ١ Print Sec 2 Search Back Netscape Print Beload Home Search Back Netscape let 👻 🕼 What's P ated let 🗸 🕼 What's Related 💰 Bookmarks 🎄 Netsite: 🜆 🖲 SMS - MTN 🖆 Channels - Tävl 📫 Search Engines Multime 🗂 Myndigheter 📑 Multimec 🕇 Myndighe Please complete the form and click Submit. Please complete the form and click Submit. nie: James Woods ne: James Woods Junior Birthdate: 1934-02-03 00:00:00. Sirthdate: 1934-02-03 00:00:00 Country: Canada untry: North Canada Submit Query Subr **a** ->-_____ 🏾 🔆 😼 🐠 🖬 🎸 Document: Done Document: Done dP 🖬 Output page for servlet Bean: test63.Update3Se /I... _ 🗆 🗙 <u>File Edit View Go</u> Communicator Help Forward Reload Home Back N alscape Search Print 🕼 Bookmarks 🙏 Netsite: vlet/test63.Update35 👻 🍘 What's Re 関 SMS - MTN 📫 Channels - Tävl 📫 Search E 🕇 Mult 🕇 Myndiaheter **UPDATE statement executed!** 🖬 ᢇ 🛛 🛛 Document: Done 🔤 💥 📲 🐠 🖬 🎸

4 ASPs

In this section we will try to create a set of ASPs to provide the same functionality as before. ASPs work in a way similar to servlets. The following figure illustrates how an ASP is executed:



Figure 3 Inside ASP plug-in - ASP execution

ASPs are very much like html files. In addition to the usual html content, an ASP includes script commands within the <% and %> delimiters. The default script language is VBScript. Everything that appears between those delimiters is executed on the server (in our case by the ASP plug-in). An ASP can look like this:

```
<HTML>
<% name = "nikos" %>
<BODY>
<B>Hej <% = name%></B>
</BODY>
</HTML>
A variable "name" with value "nikos"
A variable "name" with value "nikos"
to return the value of a variable.
```

After the script part of the ASP has been executed the ASP should look like this:

<HTML> <BODY> Hej nikos </BODY> </HTML>

4.1 Accessing a database

To access a database with ASPs is very simple. The following two lines of code establish a connection to a database:

```
<% Set db = Server.CreateObject("ADODB.Connection")
db.open "%ODBC-DSN%", "%user-name%", "%password%" %>
```

The first line creates a connection object and assigns it to the variable name "db". The second line instantiates the connection to the specific database, through an ODBC DSN. The user-name and password to the database can also be included.

At the end of the ASP the database connection can be closed:

<% db.close Set db = Nothing %>

To execute an SQL statement is also simple:

<% SQLString = "SQL statement" Set result = Server.CreateObject("ADODB.Recordset") result.Open SQLString, db, 3, 3 %>

First create a string with the SQL statement. Then create a variable to receive the result of the SQL statement. Finally run the SQL statement on a specific database connection. The third and fourth parameters of the "Open" function specify the following: *adOpenKeyset* & *adLockBatchOptimistic* In this compendium these parameters will always be set to 3.

Another method to execute SQL statements is this:

<% Set result = db.Execute("SQL statement") %>

And of course if the SQL statement's result is not important:

<% db.Execute("SQL statement") %>

To navigate through a result of an SQL statement there is the following functions:

Move to the first record: result.movefirst

Move to the last record: result.movelast

Move to the next record: result.movenext

Move to the previous record: result.moveprevious

To delete the current record in a result-set: result.delete

To check the beginning and the end of the result-set: result.bof, result.eof return true/false

Institutionen för Data-	Databases & Internet Laboration v. 1.1	Stockholm
och Systemvetenskap	*63	November 1999
SU/KTH	Relationsdatabashanteringssystem	
nikos dimitrakas		

To loop through a result-set, writing a field from each record to the web page: do until result.eof =result("FieldName") (or result.Fields("FieldName")) result.movenext

loop

To retrieve the value at a field of the current row of the result-set you can either use the field's name or its index position, starting with 0.

4.2 First ASP

Let's now take the first question and try to make an ASP for it:

List all the books (title, mainauthor, isbn) and their publisher (name and country) ordered by publisher, author and book title

To write an ASP file you just need an editor. It is recommended to use ScriptBuilder, because it supports ASP files.

- Create a new file with asp as extension. Save this file at a location were the web server can find it, for example in a directory called ASP under the web server root: c:\Netscape\SuiteSpot\docs\ASP\file.asp
- Choose what design you want for your result and write all the static html code for it. If, for example, you want to have the result of the SELECT statement in a table, you can write the html code for the table and the titles...

Here is a possible design:

Book			Publisher		
Title	Main Author	ISBN	Name	Country	
Data	Data	Data	Data	Data	
•	•	•	•	•	
•		•	•		

Add now some code at the beginning of the file to connect to the database:
<% Set db = Server.CreateObject("ADODB.Connection")</p>
db.open "lab63" %>

```
Now And some code for the SELECT statement:
  <%
  statement = "SELECT book.title, book.mainauthor, book.isbn, publisher.pname, publisher.country
  FROM book, publisher WHERE book.publisher = publisher.pname ORDER BY publisher.pname,
  book.mainauthor, book.title"
  Set resultset = Server.CreateObject("ADODB.Recordset")
  resultset.Open statement, db, 3, 3
  %>
```

Now we need to go through the result-set and populate the table. To do that we need a loop.

Institutionen för Data-	Databases & Internet Laboration v. 1.1	Stockholm
och Systemvetenskap	*63	November 1999
SU/KTH	Relationsdatabashanteringssystem	
nikos dimitrakas		

Find the part of the code that you want to repeat for every row and include it in the loop. For example:

Everything that is included in the loop is going to be the ASP result once for every row in the result-set.

- Make sure that there is matching <% and %> delimiters around the VBScript code blocks. No html content should be within the <% and %> delimiters.
- At the end of the file add some code to close the database connection:
 <% db.close</p>
 Set db = Nothing %>

When you call the ASP from a browser you should get something like this:

滋 All the books - Netscape				_ 🗆 ×
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> ommunicator <u>H</u> el	р			
🎽 🗳 🗳 🧟 🏦	🚽 🛋			N
Back Forward Reload Home Search Netsci	ape Print Security	Stop		
👔 🎲 Bookmarks 🔏 Netsite: http://localhost/asp/file.asp	a na a alla na an an	-4 5: -	▼ ♥ ₩	/hat's Related
📱 💾 SMS-MIN 🛅 Channels-Tavi 🛅 Search Engines 🦲 f	Ayndigheter 🦲 Multimedia	.,mm 🦲 Diverse 🧯	🚡 Music 📑 Help in sublect 📑 Di	verse help 🛅
Book			Publisher	
Title	Main Author	ISBN	Name	Country
Concepts of programming languages	Robert W. Sebasta	0-8053-7133-8	Addison-Wesley	USA
Sångfågeln från Milano	Annette Kullenberg	n/a	Bra böker	Sweden
Rabbit Redux	John Updike	91-37-09391-6	Forum	Sweden
Rabbit Run	John Updike	91-37-09390-8	Forum	Sweden
Management Systems Conceptual Considerations	Peter P. Schoderbek	0-256-25166-5	Irwin	US
Dödliga Data	Håkan Borgström	91-27-02803-8	Natur och kultur	Sweden
System Analysis and Design	Kenneth E Kendall	0-13-654229-8	Prentice-Hall International	UK
Vägen till C	UlfBilting	91-44-26732-0	Studentlitteratur	Sweden
The memoirs of Sherlock Holmes	Arthur Conan Doyle	n/a	Wings books	USA
The more than complete Hitchhikers Guide	Douglas Adams	0-517-69311-9	Wings books	USA
The old man and the sea	Ernest Hemingway	n/a	Wings books	USA
Document: Done			🔆 🐝 👘	🖬 🥩 //

4.3 Updating data

The only difference when updating data is that the UPDATE statement is different every time. In this section we will try to build the same structure as in section 3.3.

First we need a page that shows all the authors and lets the user select one. That can be done in the exact same way as before (with a SELECT statement). The only thing that differs is the parameters that are sent from the one ASP to the next with a form. Example:

If you have a FORM that sends the name of the author as a parameter called "aname" then the receiving ASP can access this parameter in one of the following two ways:

If the FORM used the POST method then the parameters can be accessed like this: Request.Form("aname")

If the FORM used the GET method then the parameters can be accessed like this: Request.QueryString("aname")

Another thing that may come in handy is string concatenation. The concatenation operator in VBScript is the "&" character. Example:

```
<%
fname = "nikos"
lname = "dimitrakas"
fullname = fname & " " & lname
%>
```

Complete this little application so that it behaves like the one in section 3.3.3.

5 Completed Lab requirements

The exercises in section 3 & 4 are compulsory. In addition to that, every group has to create one more set of servlets <u>or</u> ASPs for inserting or deleting data (INSERT or DELETE statement). Every group is free to choose their own statement and their own design.

Before the 22nd of December, you should do a short (oral) presentation of your work. Contact nikos to book time!

6 Internet Resources

VBScript & ASP

http://asp-help.com/

Servlets & JSP

http://www-4.ibm.com/software/webservers/appserv/doc/v20dcstd/doc/index.html

7 Epilogue

When all this is done, you should have a quite good understanding of how to use servlets and ASPs for making databases available on internet.

I hope you have enjoyed this compendium. Please come with feedback!

The Author

nikos dimitrakas