Making the Oracle6i-Oracle9i (10g) Forms Switch

Presented to

June 2004

John Jay King
King Training Resources
john@kingtraining.com

Download this paper and code examples from:
http://www.kingtraining.com

Session Objectives

- Understand how to prepare for moving client-server forms to the web
- Know the new features of Oracle iDS 9i Forms
- Be aware of the architecture of Web Forms

Introduction

- Using Developer?
- Have you heard of the Web?
- You’re in the right place
- Oracle’s Developer product now allows creation of applications that can be deployed using the Web
  (not really new, we could do this with Oracle Forms 6i)

How Did We Get Here?

IAF/IAG

SQL*Forms 2.0 & 2.3

SQL*Forms 3.0

Oracle Forms 4.0

Oracle Forms 4.5 & 5.0

Oracle iDS Forms 6i & 6i Web Forms

Oracle 9i Forms & 10g Forms

Oracle Statement of Direction

- According to Oracle's June 2004 "Statement of Direction" provided to ODTUG members
- The table below provides the specific time periods for the error correction support and extended maintenance support for Oracle Designer, Forms, and Reports 6i and 9i Releases:

Product	Error Correction Support	Extended Support
Oracle Forms, Reports and Designer 6i	01/31/2005	01/31/2008
Oracle9i Developer Suite (9.0.2) (same as Oracle 9iAS 9.0.2)	06/31/2005	06/31/2008
Oracle Developer Suite and Application Server 10g (9.0.4) and beyond	Not determined (2007 or later)	Not determined (2010 or later)

Why Web Forms?

- If you are going to Forms 9i you have no choice
- Easier deployment (maybe) and potential internet deployment
- More resilient with 9iAS server farm
- Potentially reduced network traffic
- Access to SSO and other new technology

Why NOT Web Forms?

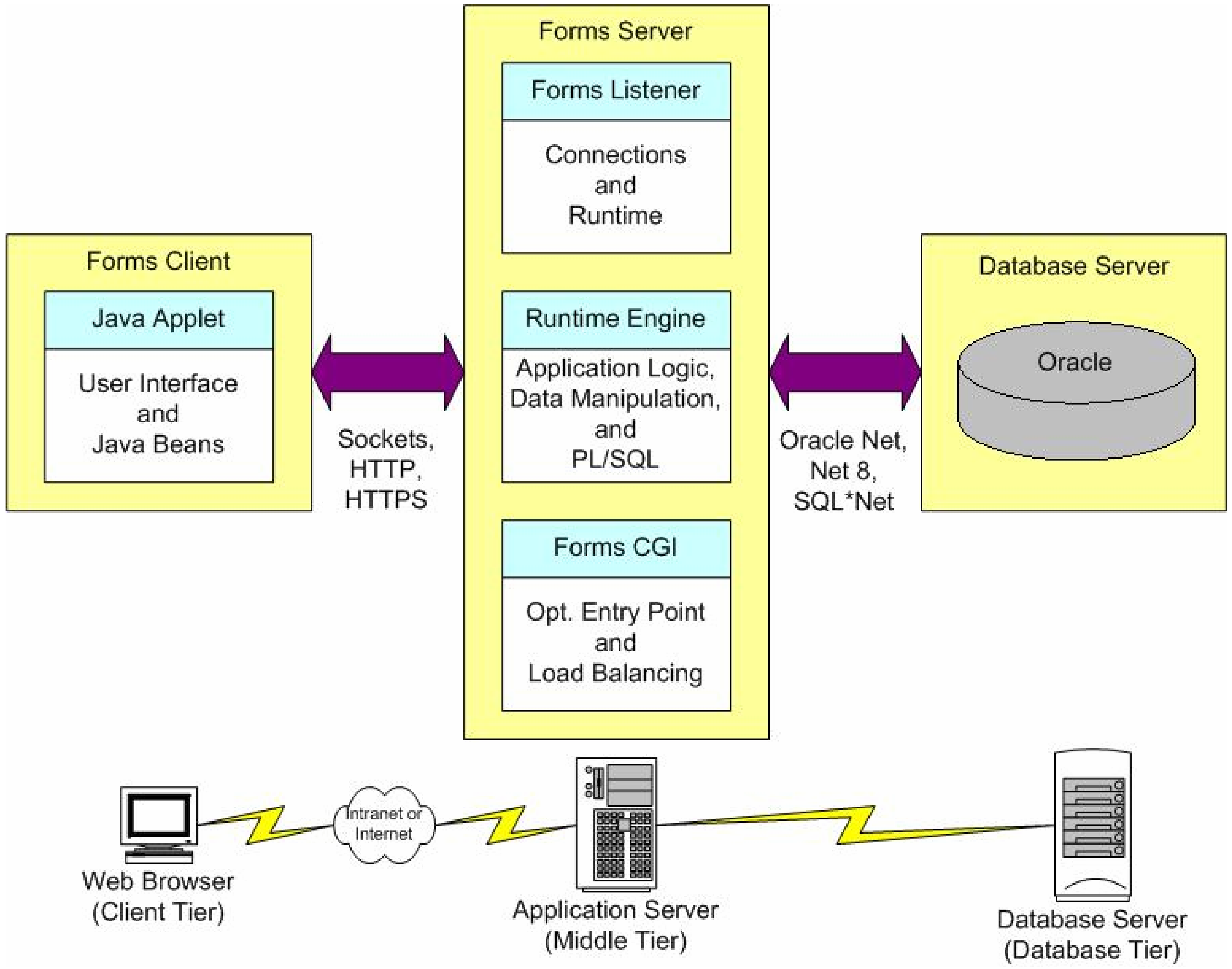
- Additional hardware required
- Additional licensing cost
- Additional complexity
- You might have applications that depend upon extremely close integration with client pc
- You might have a policy against web-deployed applications

Three-tier Implementation

- Oracle supports a three-tiered deployment environment:

– Tier 1	Client (user interface)
– Tier 2	Application Server (most application logic and business rules)
– Tier 3	Database Server (server-side application logic and data access)

Web Forms and n-Tiers



Forms Server
Forms Listener
Connections and Runtime
Runtime Engine
Application Logic, Data Manipulation, and PL/SQL
Forms CGI
Opt. Entry Point and Load Balancing
Forms Client
Java Applet
User Interface and Java Beans
Sockets, HTTP, HTTPS
Database Server
Oracle
Oracle Net, Net 8, SQL*Net
Intranet or Internet
Web Browser (Client Tier)
Application Server (Middle Tier)
Database Server (Database Tier)

Web Forms Restrictions

- Several forms features must be avoided or used differently when creating applications for the web:
  - Only some fonts are supported in both environments
  - OCX and OLE controls are not supported
  - Forms Print function is not supported
  - Some validation and triggers cause too many network trips
  - Toolbars do not always show up
    (make sure that Window Properties for Form Console and Toolbar Canvas reference CG$WINDOW1)
  - Icons must be converted
  - Graphics should be moved to HTML
  - Menu accelerator keys not functional till Java enhancement

Fonts

- Only six fonts are common between MS Windows, Motif, and Web-based forms:

Java/Web	Windows	Motif/X Windows
Courier	Courier New	adobe-courier
Dialog	MS Sans Serif	b&h-lucida
DialogInput	MS Sans Serif	b&h-lucidatypewriter
Helvetica	Arial	adobe-helvetica
Symbol	WingDings	itc-zapfdingbats
TimesRoman	Times New Roman	adobe-times

- Forms will convert fonts to whatever it thinks is closest, but, to be safe you should stick to these six fonts so that forms will appear similar when testing to what they will appear on the web

Images

- A couple of problems occur when images are used in forms that will be deployed to the web
  - Image types are not consistent between the web and client/server environments
    (web uses .gif and .jpg)
  - Images that are part of the form are rebuilt when the Java applet goes back and forth, causing unnecessary traffic and wait times
- It may be best to move boilerplate images to the html file for the application to avoid excessive traffic and waits

Icons

- The .ico format common in a client/server environment will not work on the web
- Iconic buttons on toolbars and pushbuttons must be converted to point to a web-specific image format (.gif, .jpg, etc...)
- Be careful of licensing issues here!

Mouse Events

- Some mouse events are simply not supported on the web due to excessive network traffic
- WHEN-MOUSE-ENTER, WHEN-MOUSE-LEAVE, and WHEN-MOUSE-MOVE fire as the mouse moves, this would cause a tremendous amount of network traffic if allowed
- WHEN-MOUSE-UP and WHEN-MOUSE-DOWN triggers still cause a great deal of network traffic, but, they will work

  (exercise caution for performance purposes)

Validation

- On the web form functionality takes place on the Application Server, so, network travel must occur when validation is needed
- Normally, validation is performed at the item level and the block level making the form more useful and the error messages more immediate
- Validation on the block or item level causes network traffic anytime the cursor is moved from field-to-field or that a value is changed, this may cause excessive traffic
- It may be a good idea to redesign user applications to use form level validation to reduce the network traffic, this causes less meaningful interaction with the user and may not be worth the savings

Forms Deployment

- Creating applications for web deployment means a few new steps:
  - Copy .fmx and JAR files to application server
  - If application server uses different operating system than development machine, the .fmb must be moved and recompiled on the application server
  - Run Java Applet Viewer or Browser to test the application

Deployment Issues

- The listener, DAD, and Virtual Directory need to be in place
  (Don’t forget to start and stop listener...)
- HTML needs to be modified to incorporate images and other needs
- JAR files need to be created
- Use array DML where possible
- Design forms with maximum application partitioning

Base HTML Files

- base.htm, basejini.htm, and baseie.htm are three base created by the Oracle Installer (usually they do not require modification)
- When a user first starts a Web-enabled a form, a base HTML file is read by the Forms Server
- Variables (%variablename%) in the base HTML file are replaced with the appropriate parameter values specified in the formsweb.cfg file and paramters in the URL request
- The base HTML file is downloaded to the user's Web browser
  (to customize base HTML variables, modify the parameter values in the FormsServlet.initArgs file or the formsweb.cfg file

base.htm, page 1

```
<HTML>
<!-- FILE: base.htm (Oracle Forms)                          -->
<!--                                                        -->
<!-- This is the default base HTML file for running a form on the   -->
<!-- web using a generic APPLET tag to include Forms applet.        -->
<!--                                                        -->
<!-- IMPORTANT NOTES:                                       -->
<!-- Default values for all the variables which appear below        -->
<!-- (enclosed in percent characters) are defined in the servlet    -->
<!-- configuration file (formsweb.cfg). It is preferable to make    -->
<!-- changes in that file where possible, rather than this one.     -->
<!--                                                        -->
<!-- This file will be REPLACED if you reinstall Oracle Forms, so   -->
<!-- you are advised to make your own version if you want to make   -->
<!-- want to make any modifications.  You should then set the       -->
<!-- baseHTML parameter in the Forms Servlet configuration file     -->
<!-- (formsweb.cfg) to point to your new file instead of this one.  -->
<HEAD><TITLE>%pageTitle%</TITLE></HEAD>
<BODY %HTMLbodyAttrs%>
%HTMLbeforeForm%
```

base.htm, page 2

```
<!-- Forms applet definition (start) -->
<APPLET CODEBASE="%codebase%"
        CODE="oracle.forms.engine.Main“  ARCHIVE="%archive%"
        WIDTH="%Width%“  HEIGHT="%Height%">
<PARAM NAME="serverURL" VALUE="%serverURL%">
<PARAM NAME="networkRetries" VALUE="%networkRetries%">
<PARAM NAME="serverArgs"
       VALUE="module=%form% userid=%userid% sso_userid=%sso_userid%
      %otherParams%">
<PARAM NAME="separateFrame" VALUE="%separateFrame%">
<PARAM NAME="splashScreen"  VALUE="%splashScreen%">
<PARAM NAME="background"  VALUE="%background%">
<PARAM NAME="lookAndFeel"  VALUE="%lookAndFeel%">
<PARAM NAME="colorScheme"  VALUE="%colorScheme%">
<PARAM NAME="serverApp" VALUE="%serverApp%">
<PARAM NAME="logo" VALUE="%logo%">
<PARAM NAME="imageBase" VALUE="%imageBase%">
<PARAM NAME="formsMessageListener" VALUE="%formsMessageListener%">
<PARAM NAME="recordFileName" VALUE="%recordFileName%">
</APPLET>
<!-- Forms applet definition (end) -->
%HTMLafterForm%
</BODY>  </HTML>
```

formsweb.cfg, page 1

```
# $Id: formsweb.cfg,v 1.23 2002/01/25 06:51:41 oraforms Exp $
# ----------------------------------------------------------
# formsweb.cfg - Forms Servlet default configuration file
# ----------------------------------------------------------
# This file defines parameter values used by the FormsServlet (f90servlet)

# ********************************
# DEFAULT CONFIGURATIONS
# ********************************
#
# These are the default settings.  Any of them may be overridden in the
# Named Configurations section.  If they are not overridden, then the
# values here will be used.
# System Paremeters cannot be overridden in the URL.  User Parameters can.
#
#
# SYSTEM PARAMETERS
# -----------------
# These have fixed names and give information required by the Forms
# Servlet in order to function.  They cannot be specified in the URL query
# string.  But they can be overriden in a named configuration (see below).
# Some parameters specify file names: if the full path is not given,
# they are assumed to be in the same directory as this file.  If a path
# is given, then it should be a physical path, not a URL.
#
```

formsweb.cfg, page 2

```
baseHTML=base.htm
baseHTMLjinitiator=basejini.htm
baseHTMLjpi=basejpi.htm
baseHTMLie=baseie.htm
HTMLdelimiter=%
#
# WorkingDirectory defaults to <oracle_home>/forms90 if unset.
#
workingDirectory=
envFile=default.env
#
# The next parameter specifies how to execute the Forms applet under
# Microsoft Internet Explorer 5.x.  Put IE=native if you want the
# Forms applet to run in the browser's native JVM.
#
IE=JInitiator
#
# USER PARAMETERS
# ---------------
# These match variables (e.g. %form%) in the baseHTML file. Their values
# may be overridden by specifying them in the URL query string
# (e.g.
   "http://myhost.mydomain.com/servlet/f90servlet?form=myform&width=700")
# or by overriding them in a specific, named configuration (see below)
#
```

formsweb.cfg, page 3

```
#
# 1) Runform arguments:
#
form=test.fmx
userid=
#
# These settings support running and debugging a form from the Builder:
#
otherparams=debug=%debug% buffer_records=%buffer%
   debug_messages=%debug_messages% array=%array% query_only=%query_only%
   quiet=%quiet% render=%render% host=%host% port=%port% record=%record%
   tracegroup=%tracegroup% log=%log% term=%term%
debug=no
buffer=no
debug_messages=no
array=no
query_only=no
quiet=yes
render=no
host=
port=
record=
tracegroup=
log=
term=
```

formsweb.cfg, page 4

```
#
# 2) HTML page title, attributes for the BODY tag, and HTML to add before and
#    after the form:
#
pageTitle=Oracle9iAS Forms Services
HTMLbodyAttrs=
HTMLbeforeForm=
HTMLafterForm=
#
# 3) Values for the Forms applet parameters:
#
serverURL=/forms90/l90servlet
codebase=/forms90/java
imageBase=DocumentBase
width=650
height=500
separateFrame=false
splashScreen=
background=
lookAndFeel=Oracle
colorScheme=teal
logo=
formsMessageListener=
recordFileName=
serverApp=default
```

default.env

- The default.env file has environment variables including some attached to specific applications
- ORACLE_HOME=C:\oracle\ora92ids

```
#
# Search path for Forms applications (.fmx files, PL/SQL libraries)
# If you need to include more than one directory, they should be semi-colon
# separated (e.g. /private/dir1;/private/dir2)
#
# FORMS90_PATH=C:\oracle\ora92ids/forms90

#
# The PATH setting is required in order to pick up the JVM (jvm.dll).
# The Forms runtime executable and dll's are assumed to be in
# C:\oracle\ora92ids\bin if they are not in the PATH.
# In addition, if you are running Graphics applications, you will need
# to append the following to the path (where <Graphics Oracle Home> should
# be replaced with the actual location of your Graphics 6i oracle_home):
#
# ;<Graphics Oracle Home>\bin;<Graphics Oracle Home>\jdk\bin
#

PATH=C:\oracle\ora92ids\bin;C:\oracle\ora92ids\jdk\jre\bin\classic
```

registry.dat

- Registry.dat has paths to various libraries

```
# appFontname  represents a comma delimited list of Application Font Names.
# javaFontname represents a comma delimited list of Java Font Names.
#
# The number of entries in the appFontname list should match the number in
# the javaFontname list.  The elements of the list are comma separated and
# *all* characters are taken literally, leading and trailing spaces are
# stripped from Face names.
#
# Note that this file uses the Java 1.1 Font names in order to be able to
# handle the NLS Plane (BUG #431051)
#
default.fontMap.appFontnames=Courier
   New,Courier,courier,System,Terminal,Fixed,Fixedsys,Times,Times New
   Roman,MS Sans Serif,Arial
#
# The Application Level icon files are relative to the DOCUMENTBASE
#   example: icons/
# or an absolute URL.
#   example: http://www.forms.net/~luser/d2k_project/
#
default.icons.iconpath=
default.icons.iconextension=gif
```

Web Forms Execution

- The sequence of events supporting Web Forms Execution is as follows:

1. User accesses the URL of an HTML page that indicates a Forms application should be run
2. The HTML page and any associated JAR file(s) are downloaded to the Web browser
3. Forms applet is started, parameters from the HTML page indicate which form to run
4. Forms applet sends a request to the Forms Listener
5. Forms Listener connects to a Forms Runtime Engine process (may pass runtime parameters)
6. Forms Listener establishes connection with the Forms Runtime Engine and sends the connection information to the Forms applet
7. Forms applet connects to the Forms Runtime Engine directly (Forms Listener is freed to accept requests from other users)
8. Forms Applet displays the form's user interface using the user's browser
9. The form running in the Forms Runtime Engine communicates with the database

Moving from 6i to 9i Forms

- Removal of client/server runtime
- Removal of character mode
- Impact of client/server removal
- De-support of legacy built-ins
- Stricter enforcement of existing triggers and built-ins
- Database no longer available for module storage
- Mouse trigger changes
- Legacy menu desupport

9i Web Forms New Features, 1

- All forms are web-only, client-server forms no longer supported
- Servlet Architecture provided by the Forms Listener Servlet provides the ability to run forms from any web browser
- Single Sign-on Support using get_application_property(sso_userid) support organization standard sign-on capability
- DATETIME TIMEZONE properties FORMS90_DATETIME_LOCAL_TZ FORMS90_DATETIME_SERVER_TZ FORMS90_TZFILE take advantage of Oracle9i Date and Timezone capabilities
- JDAPI (Java Design-time API) provides Java2 API used to create, JDAPT uses the Forms C API and lets Java programmers work with Forms without needing to learn C (compile-time only, not runtime)

9i Web Forms New Features, 2

- Oracle Containers for Java2 Enterprise Edition (OC4J) provides a complete J2EE (Java2 Enterprise Edition) server that contains a Java Servlet engine (used by Forms as the default) and other features:

  Each HTTP request is received by the Oracle HTTP Listener, passed to the OC4J Forms Listener Servlet, to create a Forms Server Runtime process that manages communications between the client browser and the runtime engine
- Oracle9i Forms module files may be saved as (.fmb, .mmb, and .olb files) to XML, providing "human-readable" and XML-tool manipulatable documents
- Long running List of Values objects now display the number of LOV values processed so far and allow cancellation prior to completion of the LOV's data

9i Web Forms New Features, 3

- One-time WHERE clauses allow changing of WHERE clause values for the next execution only
- Dynamic Javabean components allow deployment of Javabeans within forms without creation of "wrappers"
- The Oracle9i Forms Trace allows tracking of record information for a specific part of a form via definition of user events
- Oracle9i has a Forms Debugger allowing complete testing of PL/SQL code both locally and remotely

What's Gone with 9i Forms

- Oracle forms listener
- Oracle procedure builder
- Oracle project builder
- Oracle translation builder
- Oracle terminal
- Open client adapter
- PECS
- PVCS & Clearcase integration

Upgrading 6i to 9i

- Upgrading from 6i to 9i
  - Forms Builder
  - Forms Compiler
  - Forms JDAPI (Java API)
- Forms Migration Assistant
  - A java based tool that warns of potential upgrade issues and can make modifications to the code.
- Third-Party Tools
- Batch

```
ifcmp90.exe file=<filename> userid=<user/pass> upgrade=yes
```

Oracle No-No's

- Oracle wants you to stop using:
  - User Exits/Call Interface
  - Executables
  - Command line options
  - Others

Tuning

- Database access
  - Optimize SQL access
  - Array fetches, etc
- Application tuning
  - Good coding practice
  - Efficient code, etc
  - Lots of advice out there
- Client hardware
  - The more memory and CPU, the better
- Network Tuning
- Oracle Application Server Tuning

Reduce Items on Screen

- Objects that display on a screen need to be initialized
- Less items on an initial screen speeds up performance
  - Have a “skinny” first screen
  - Hide items that are not needed
  - Avoid tabbed canvases unless download deferred
    - WHEN-TAB-PAGE-CHANGED trigger
    - RAISE_ON_ENTRY=YES
    - VISIBLE=NO

Reduce Round-Trips, page 1

- Design applications that do not require a user to navigate through fields if default values are accepted
- SYNCHRONIZE Built-In is probably not good…
  - This tool often used to improve client-server problems causes performance hits on the Web
  - Roundtrip from server to client
  - Overuse generates unnecessary network traffic
  - Problems handled by "synchronize" on client/server may no longer exist on the Web

Reduce Round-Trips, page 2

- Timers
  - Forms Client maintains the timers
  - Each timer that fires results in a network roundtrip
  - Example
    - 1 timer per second for 500 concurrent users…
    - 500 roundtrips per second
- Don't abuse timers
  - Reduce frequency where possible
  - Remove timers that represent a clock (Java Bean instead)
  - Use animated images on buttons where animations are required and currently handled by timers
- Make sure that timers are stopped when no longer needed

Reduce Round-Trips, page 3

- Convert boilerplate labels to prompts
- Replace boilerplate arcs, circles and polygons with rectangles and lines
- Iconic buttons on the Web are in gif and jpeg formats
- Store application images in a single .jar file
  - Jar files are permanently cached on the client
  - Copy myimages.jar to forms90/java
  - formsweb.cfg configuration
    - Imagebase=codebase
    - archive_jini=f90all_jinit.jar, myimages.jar

Reduce Round-Trips, page 4

- Store Pluggable Java Components (PJCs) and Java Bean sources in .jar files
  - Java archives are permanently cached on the client
  - Download only when new or modified
  - Separate Java classes that need to be signed from classes that don't need security
  - Java on the client adds to memory footprint
- “Fake” startup speed
  - User satisfaction about startup time has much to to with impression
  - Use a splash screen to give the illusion of application starting
  - If using a custom splash screen image put it in the image .jar

Wrapping it all Up

- Oracle 9iDS Web Forms are the wave of the Oracle Developer future
- Modifications in functionality are required for web forms deployment (user expectations may need some work…)
- Organizations should consider moving 6i Client-Server to Web Forms as soon as possible
- Organizations with 6i Web Forms should migrate to 9i or 10g forms

RMOUG
ROCKY MOUNTAIN ORACLE USERS GROUP

Training Days 2005

Mark your calendar for:

February 9-10, 2005!

King Training Resources

To contact the author:

John King

King Training Resources

6341 South Williams Street

Littleton, CO 80121-2627 USA

1.800.252.0652 - 1.303.798.5727

Email: john@kingtraining.com

Thanks for your attention!

Today's slides and examples are on the web:

http://www.kingtraining.com