



# ORACLE®

---

## FUSION MIDDLEWARE

# 12<sup>c</sup>

### FORMS

## All roads lead to Forms 12c

Frank Hoffmann, Cologne Data GmbH

On 23 October 2015, Michael Ferrante, Principal Project Manager responsible for Oracle Forms, used his Twitter account, “@OracleFormsPM”, to announce Version 12c of Forms. The announcement marked the end of a long period of waiting, and facts took the place of many rumors.

Oracle Forms has a future – at least until 2025. Despite vague rumors to the contrary, Reports 12c has also been launched and will survive for another full application generation. Users who had regularly kept Oracle supplied with suggestions for the new Forms version up to the beginning of the year now have seen some of these suggestions implemented. Forms 12c has nevertheless proved to be a conservative functional upgrade from its predecessor, Forms 11gR2. Important libraries (“webutil.pll”) and functions have been carried over unchanged, Forms 12c documents copied from existing 11gR2 documents, and the functionality of Reports has not changed since Version 10g.

At this year’s DOAG conference, Michael Ferrante delivered a helpful presenta-

tion of Forms 12c’s new features. The main changes described in this article have been made under the hood. The article also draws upon many years of experience gained in migration and will provide decision-makers with convincing arguments for upgrading to Forms 12c.

### The history

In the mid-1990s, many Oracle customers opted for Forms as a way of developing their database applications. The choice was also recommended and promoted at the time by Oracle Consulting. Release 6iR2, in which two preceding GUI versions (4 and 5) came to

maturity, was able to enter production directly. Sadly, the wave of enthusiasm accompanying this development has since dissipated, and talk on the market is now dominated by Apex, ADF and pure Java applications. Projects doubtless exist that can be implemented with these tools and that deliver impressive results with dynamic HTML and JavaScript.

Many older – but not out-of-date – development managers, of whom the author considers himself one, still maintain however that there is no better and more productive tool than Oracle Forms for projects involving complex database models and workflows. This being the case, why has Oracle not been as strong an advocate as it was in the 1990s? Why is greater preference generally given to

|                                    | Forms 6iR<br> | Forms 11gR<br> | Forms 12c<br>   |
|------------------------------------|--|---|--|
| Date of publication                | Summer, 2000   | January, 2011   | October, 2015  |
| Support (extended)                 | 2005 (2008)  | 2016 (2018)   | 2020 (2023)  |
| JDK 32-Bit                         | JDK 1.1  | JDK 1.6   |  |
| JDK 64-Bit                         | -  | JDK 1.7   | JDK 1.8  |
| Client environment                 | Installation runtime version   | JRE 1.7.40+<br>JRE 1.8.51+  | JRE 1.7.55+<br>JRE 1.8.51+   |
| PL/SQL version                     | 8.0.6.3  | 11.1.0.7  | 11.2.0.3   |
| Database                           | runs<br>„unsupported“ at 8.11g and 12c<br>with sqlnet adjustment                               | 10.2,<br>11gR1 R2,<br>12c R1  | 11.2.0.4, 12.1.0.2<br>+ database repository for each<br>installation (DEV0..DEV99)                 |
| Certificated browsers              | -  | Windows<br>IE 8,9,10,11 FF24+ MAC IOS<br>Safari 7+, 8+  | Windows<br>IE 11<br>FF 31+<br>MAC IOS (*)<br>Safari 8+   |
| SQLNET                             | 8  | 11  | 11   |
| Forms/Reports<br>client connection | via SQLNET 8 (client/server)   | http/https from browser with<br>JRE 1.7,1.8 plug-in   | http/https via JRE 1.8 or<br>webstart (JNLP) or<br>JRE stand alone<br>(without JavaScript and SSO) |
| accessibility                      | no   | Java Access Bridge with<br>JAWS12+  | Java Access Bridge with<br>JAWS12+   |

Table 1 (\*) a test of Forms 12C with a topical MacBook has worked well. After installation of JRE 1. 8. 66 for Mac the Applet is begun. The Java process of the Forms application was indicated in Mac even at the term in the task bar and was finished few seconds after closing the browser.

pure Java and HTML-based solutions? Why were there only three talks on Forms at the DOAG conference – and not one of them by Oracle Germany? No one really knows. One thing is certain though: it needn't stay that way in the years ahead.

## The fascination of Forms

As its name implies, Oracle Forms is a highly productive tool for the creation of forms. Once the development logic of Forms has been grasped, phenomenal results can be obtained with PL/SQL and a few development rules. The logic and the basic functionality of triggers, procedures and parameters can be learned relatively easily. Provided programming is structured, the database is kept logical and rules are followed during building of the application (style guide, notation, inline documentation, structured development, etc.), the result is highly

sustainable software. You could start a new project with Oracle Forms 12 today.

Table 1 shows the main developments. The information has been taken from the certification tables and Oracle documentation as at 9 December 2015.

Forms 12c relies upon a new generation of databases and browsers. Java processes run with greater stability than in the preceding version. The highlight of the new version for the author is the facility for standalone access – i.e. without the need for a browser.

## Forms 12c server architecture

For demonstration purposes, Forms/Reports 12c was installed on a Windows 2012 R2 server. The HTML pages were to be displayed dynamically by “mod\_owa” (open source). Oracle 12c Standard Edi-

tion was selected as the database, and an SSL certificate installed for access via SSL. Users should check that their chosen certificate provider is also on the Oracle Java CA list, since browsers such as Firefox and Safari otherwise report the address as untrusted (even when no self-signed JAR files are installed). An Apache 2.4 reverse proxy routes the connections to Forms in this configuration. In the installation, Reports runs in the background using BLOBdestination and generates PDF files in the database (see Figure 1 and Listing 1).

Security was further enhanced by a gateway form. This form is able to perform a number of security checks and to check the passwords of the destination database by global links before a Forms session is launched against a production database (see Figures 2 and 3).

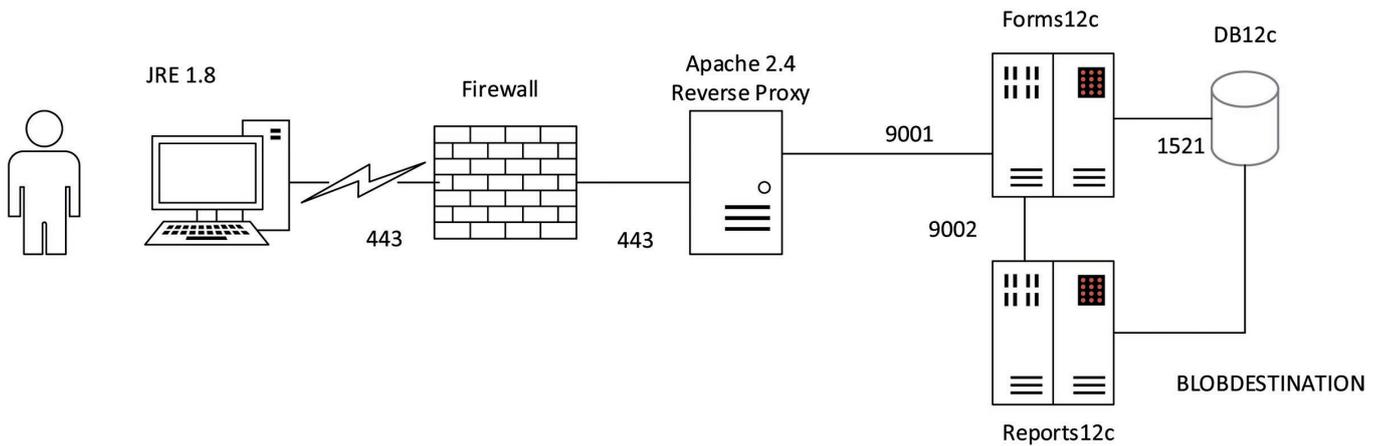


Figure 1: Form 12c architecture with a coded https connection

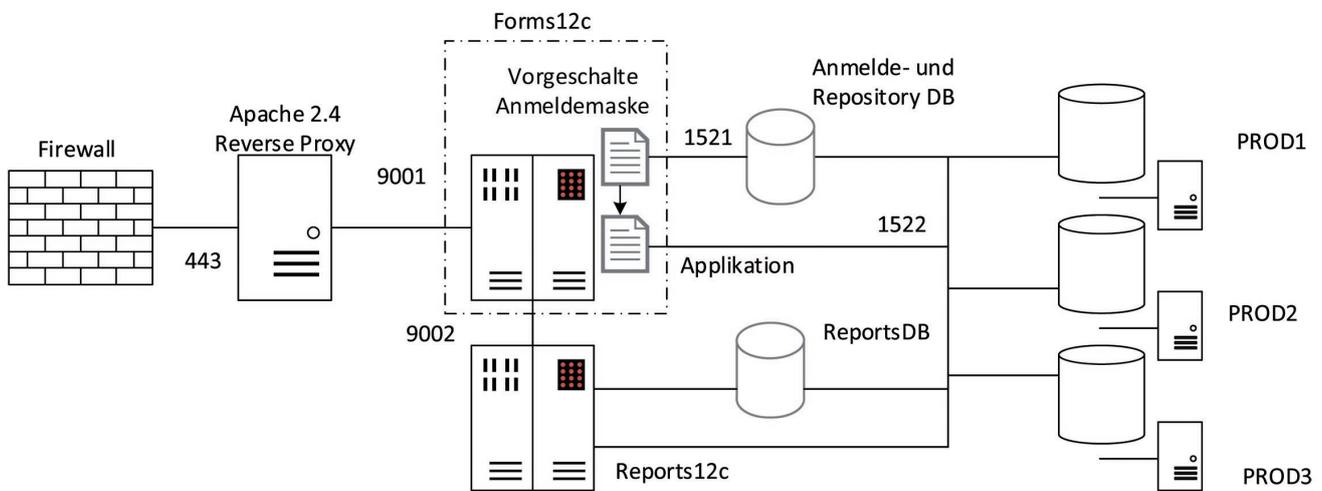


Figure 2: Raised security with added login form

## Login database with login form

When the application is launched, the login form automatically connects to the login database. This enables an invalid login to be logged before a login to the production database is attempted. When the password is entered, the login database uses a global link to the destination database to check the password (the alias "PROTEST" for example links to "PROD1"). If the password is correct, the "NEW" form connects to "PROD1" through Port 1522. If the password is incorrect, the user is rejected.

The login database can also be used as a central location for the new Forms 12c repository. A second central database for storing all asynchronous reports is also conceivable. The advantage here is that the production databases are not encumbered unnecessarily with configuration schemes from Forms 12c and

large BLOB columns from Reports, overloading the online backups. Security requirements can be covered as follows:

- Defense against denial-of-service attacks: should the form be flooded by a DOS IP, it is terminated immediately, and a warning message is sent by e-mail
- Security delays in the event of multiple login attempts
- No more than one login window permitted per client ID

- Login with the same login name from multiple computers is not permitted
- Critical Java and browser versions are rejected
- Security warnings are sent by e-mail in the event of logins at unusual times

```
<VirtualHost colognedata.com:443>
ProxyPreserveHost On
ProxyRequests Off
ServerName colognedata.com
ProxyPass /forms/ http://127.0.0.1:9001/forms/
ProxyPassReverse /forms/ http://127.0.0.1:9001/forms/
</VirtualHost>
```

Listing 1: Reverse proxy solution to start Forms

|              |                          |           |
|--------------|--------------------------|-----------|
| Benutzer(in) | <input type="text"/>     | Anmeldung |
| Passwort     | <input type="password"/> |           |
| Datenbank    | PROTEST                  |           |

Figure 3: Login

## An introduction to the world of Forms 12c for autonomous learners

Besides the talk by Michael Ferrante at the DOAG 2015 conference, the Oracle page ["http://docs.oracle.com/middleware/1221/formsandreports/index.html"](http://docs.oracle.com/middleware/1221/formsandreports/index.html) (see Figure 4) provides a good introduction to the use of Forms 12c.

The user is advised to study certain further documents concerning the system requirements before conducting an installation and the first tests:

See: ["http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-requirements-100147.html"](http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-requirements-100147.html)

Product certifications:

See: ["http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html"](http://www.oracle.com/technetwork/middleware/ias/downloads/fusion-certification-100350.html)

Forms community and support for 12c bugs:

See: ["https://community.oracle.com/community/development\\_tools/forms/content"](https://community.oracle.com/community/development_tools/forms/content)

## Conflicts in Forms 12c

Users wishing to keep their forms and reports in one directory and to configure this directory in the "formsweb.cfg" configuration file can use the "WorkingDirectory" applet variable for this purpose. This creates a shared runtime directory for a configuration. In Forms 12c, the "FORMS\_MODULE\_PATH" variable must however then be commented out in "default.env" (see Listing 2).

## New event types in Forms 12c

Forms 12c is now also able to communicate with the outside world through new system events. New inactivity timers can be defined in Forms 12c and polled by triggers. In the past, this was possible only with Java Beans. For example: the new inactivity timer is set to 30 seconds after accessing of the form with "SET\_APPLICATION\_PROPERTY(CLIENT\_IDLE\_TIME,30);". An event of the type "System Client-Idle" is then defined and a "WHEN-EVENT-RAISED" trigger created with the command: "message(Client idle trigger trips , || :system.last\_event);". The result: after thirty seconds of client inactivity, the trigger is tripped.

Figure 4: Oracle homepage

## Launching Forms 12c as a standalone Java applet without a browser

This feature is both very useful and easy to implement. It results however in the SSO and JavaScript features being lost. For the feature to be used, the .jar file "frmsal.jar" must first be installed on the client, for example from the author's demonstration server (see ["http://forms12c.com/forms/html/fsal.htm"](http://forms12c.com/forms/html/fsal.htm)). A local batch file for launching Forms (see Listing 3) can then be created. In this case, "frmsal.jar" is located in the "C:\oracle\java" directory. A launch log then appears in the DOS window (see Listing 4). The DOS window is closed automatically when the application is closed.

```
Default.env:
#FORMS_MODULE_PATH=%FORMS_PATH%
```

Listing 2: Adjustment of the „default.env“

```
c:
cd C:\oracle\java
java -jar frmsal.jar -url "http://forms12c.com/forms/frmservlet?config=s
tandaloneapp" -t 30000
```

Listing 3: Standalone Java applet start

## Latency measurements and network analyses in Forms 12c

As previously in Forms 11gR2, latency measurements can be performed by the addition of environment variables (see Listing 5). The parameters must also be entered in the corresponding "jpi" document.

The examples can be accessed in http or https on "http://forms12c.de/demo". The page also contains the links and documents referred to here and a copy of the talk on Forms 12c given by Michael Ferrante at the DOAG conference. In this example, the latency test is started by a double click on the roundrips display.

## Reading icons from a directory without a ".jar" file

Icons can be read in from a launched 11gR2 or 12c session at runtime. This was tested successfully in Forms 12c with both the browser and standalone versions (see Listing 6). For this purpose, two entries must be made and the WebLogic server restarted. A restart is also necessary after any future change in the icon directory, owing to the cache entries. A new "icons" directory containing the relevant GIF files must also be created under "webutil".

## Migrating from older versions of Forms

Users of a current Oracle Forms version are familiar with the problem of browsers and Java plugins. Security mechanisms (JavaScript) must be integrated in order to prevent "Forms browser"

windows from being closed inadvertently. Browsers such as Firefox and Internet Explorer have their idiosyncrasies. As can be seen from the table, certain browsers are certified and others not in each version. A further problem is the purging of the local client Java sessions (zombie processes) following a browser crash or dropping of the connection. These processes can render the system unstable. Often, the only solution for the user is to log out of the client system. Some of these bugs in 11g have been fixed in 12c. Migration is limited to a batch recompile of all modules. The benefits of Forms 12c compared to 11gR2 are:

- *Browser plugins are no longer used to access forms*  
The new standalone solution in Forms 12c enables the application to be run entirely without a browser in a Java Runtime Environment (JRE). Unfortunately, this comes at the price of the loss of certain options: SSO and JavaScript are then no longer supported.

- *Support through 2020*  
At the end of 2016, support for 11gR2 expires and the fee-based "extended support" begins. Furthermore, Mozilla has announced that from the end of 2016, Java plugins will no longer be permitted in Firefox. Users must also check whether the patch for the .jar files has been applied to 11gR2; the certificates otherwise expire at the end of January 2016.

- *Optimized JVM memory management of Forms and Reports*  
A major problem of Version 11gR2 is the JVM memory management. Bug fixes and optimizations in the new version hold the prospect of improvements in this area. The new version also moves record management of a Forms block into the memory. With the optimized JVM memory method, the report server no lon-

ger needs to be restarted regularly.

- *Integration of BI-Publisher*  
New products such as BI-Publisher can be integrated in a similar way to Reports.
- *Installation of WebUtil without OLE2*  
For users who do not need OLE2, Forms 12c permits installation completely free of self-signed .jar files ("webutil\_no\_ole"). The supplementary .jar file for automated integration of MS Office, which before now had to be self-signed ("jacob.jar"), is no longer required in this case.
- *New system event types for inactivity monitoring*  
The "timeout.jar" file can be replaced by new Forms 12c event types. This reduces development time and eliminates self-signing issues with an expensive code-signing certificate.

## Technical arguments for an upgrade from 6i to 12c

Only Version 8 of PL/SQL can be used for programming in the environment of 6i, which dates back to the year 2000. More recent PL/SQL functions are not supported. The same applies to the C compiler: legacy C code for interfaces can be maintained only with extremely old compilers. The JDK 1.1 distributed with Forms 6i presents a security risk, and its functionality is also that of the year 2000. Integration of the database with "NET8" is obsolete. Connection to Version 12 of the database requires modification of "sqlnet.ora". The arguments for abandoning Forms 6i completely are as follows:

```
Archive files are downloaded in the directory C:\Users\COLOGN~1\AppData\Local\Temp\frmsal\12.2.1.0
Forms-Session-ID: WLS_FORMS.formsappp.51
Proxyhost: null, Proxyport: 0.
Native HTTP implementing is used for connection.
Connecting mode: HTTP.
Version of Forms-Applet: 12.2.1.0
```

Listing 4

- *The continued use of Forms 6i constitutes a security risk*

The use of Forms 6i on the intranet and over the Internet requires an open database port. This makes the database susceptible to attack. Hacking and abuse of the database accounts are possible directly over the Internet. The required Version 8 of SQLNET offers little in the way of security functions and dates back to 2000. As with SQL code, passwords are generally transmitted unencrypted. Alexander Kornbrust, an expert in the field of database security, demonstrated the scale of the security risk presented by an open database port some time ago and with numerous examples. The risk is high on the intranet and catastrophic on the Internet. The entire database traffic is transmitted down the lines and can read with ease.

- *A local database installation such as that in Forms 6i reveals too much*

Forms 6i requires a runtime installation with NET8. A party aware of the database connection can use 3rd-party tools to create an undesired database connection. A malicious party can easily launch a DOS attack with the host address. The party always knows the port and host address of the database to which he logs in. This is difficult to prevent.

- *Mixed-case passwords are not possible*

Since Oracle transfers passwords to Reports in upper case, passwords using a combination of upper and lower case are not possible. This further impairs security. The passwords used in the database must therefore take account of Forms 6i, which in turn reduces the security offered by the choice of password.

- *Possible overloading of the network with 6i modules*

Reports for bulk printing or long lists may access the database multiple thousands of times. These operations access the database and the network line by line and for hours at a time. In some forms, CLOB or BLOB columns are transferred by Forms line by line to the client. Upgrading to Forms 11g/12c allows for a considerable improvement here. Firstly, the network connections between the database and the report server are short and secure. Secondly, the reports can run asynchronously in the background on the application server, and do not block the client. Forms modules with high database traffic also run faster on the new Forms versions.

# Mit dem **DOAG EX aday** zur *Pole-Position*

*13. April 2016 in Hamburg*



```
Formsweb.cfg:
latencyCheck=true      # latency measurement of 1. 4 kb and 64 kb packages
networkStats=true     # display roundtrips and transference volume
```

Listing 5: Latency measurement with Forms 12c

```
Formsweb.cfg: imageBase=documentBase
Registry.dat: default.icons.iconpath=webutil/icons/
```

Listing 6: Icons import from "webutil"-directory

- **Security risk with distributed modules**  
The Forms modules for operation of an application can be interchanged as desired. The risk of abuse here should not be underestimated. The authenticity of the application cannot be assured. It is sufficient to change the access paths of the modules in order to infiltrate the system with manipulated modules.

## Migration from 6i to 12c

An overview of possible changes and adaptations can be found in the guide to upgrading from Forms 6i to 12c ([see "http://docs.oracle.com/middleware/1221/formsandreports/upgrade-forms/index.html"](http://docs.oracle.com/middleware/1221/formsandreports/upgrade-forms/index.html)). The guide has hardly changed from the version for 11gR2. The comments made in earlier articles on 11gR2 are therefore for the most part still valid. In 2012, the author prophetically entitled his DOAG talk "Experience gained with migration from Forms/Reports 6i to Forms/Reports 11.1.2 and 12". The content of this talk still stands; only the "c" suffix was not anticipated at that time.

At the start of any given project, the legacy 6i code should be analyzed, either by batch export of the modules to XML or text, or by smart API tools. Critical search terms such as "HOST", "OLE2", "FFI", "RUN\_PRODUCT" and "TEXT\_IO" assist in identifying forms that could present problems. In principle, all forms with such calls can be migrated automatically; calls on 11g/12c must however be tested thoroughly following migration.

WebUtil functions always need a visible "canvas" in order to work; some menu calls may require auxiliary forms with a "webutil" canvas. Numerous variants could be programmed on Forms 6i that run differently on 11gR2, if at all. On

Unix, Windows fonts may have to be installed; reports may otherwise appear with spurious characters on Unix.

If OLE2 is used, a signed "jacob.jar" file must be integrated. It is worthwhile developing a concept for integrating new 12c logic automatically and in a standardized way during migration. Triggers, libraries and comments should be added consistently in all modules. Some degree of graphics work will be necessary: all icons must be converted to .gif files with transparent backgrounds. Logos could be adapted at the same time and 16x16 pixel images created for the new tab label icons function in Forms 12c.

MDI toolbar and trees require an icon that is aligned top left. All other buttons require centered icons, ideally of 32x32 pixels. When an icon is used twice (for MDI toolbar and form), two versions of the icon must be created. The use of 6i and 12c in parallel is recommended in order to avoid risks of production failure. It may however be advantageous for further development to be frozen, as testing of the new version will otherwise always be piecemeal and difficult, since it can never be based on a definitive version – and may never be completed.

Configuration and deployment should not be underestimated; implementation with WebLogic servers is resource-intensive, for both client and server. If reports are to be developed further, the attractive Forms 12c standalone solution of the Forms 12c builder is not an option, and as in 11gR2, the result is cumbersome and resource-intensive installations with at least 4 GB of additional main memory.

The lines to the forms server must be tested for latency in access tests. In security networks, the ability of the local DNS server to resolve the certificates of the Java applets must be checked. 64-bit

Java clients may have to be excluded, since they could be unstable. Oracle merely recommends switching to 32-bit Java versions in the event of problems with 64-bit runtime environments.

Not every frame and every pixel interval can be transferred unmodified to the new version. Minor manual adaptations will be subsequently be required. A future goal in 12c will continue to be reducing the network loads. It may on occasions be advantageous to generate a document on the server first and then to send it to the client in full. Good documentation assists in making the process of migrating a module more readily comprehensible at a later stage.

## Reports 12c

Two points in brief: Reports has not been changed since Version 10g, and is no longer receiving particular attention from Oracle. Oracle is not willing to say whether a Reports 13c will appear. Reports is at any rate no longer being developed. We can assume that it will continue to stagnate. The good news however is that the migration from Reports 6i to Reports 12c involves very little effort.



Frank Hoffmann  
frank.hoffmann@cologne-data.de  
<http://forms12c.de/demo>  
@forms12c