MainTrend Ltd.

E-mail: info@maintrend.net
Site: www.maintrend.net

Rally to Java (J2EE) Thin Client Migration

White Paper

November 2008

Have you ever asked yourself: "What I am going to do with my great unsupported Oracle Rally applications?"

If so, this document is for you.

Contents

Executive Information

This document describes MainTrend’s Solution for Rally to Java Conversion. Its purpose is to provide general information and understanding of the conversion process and JDataPanel Framework as a basis for this solution and a typical, general methodology that can be applied to application conversion projects utilizing JDataPanel Framework.

Rally 4GL for rapid application development was adopted by many enterprises for their vertical application development in the late 1980's and early 1990's. In the context for which it was created, Rally was a formidable tool that allowed development of applications in the Open VMS environment for text-based systems. Many enterprises and organizations built their mission critical systems with Rally. Many man-years were invested in Rally code, in the application user interface and business logic.

But now Rally applications have become legacy ones: they have many limitations and do not meet modern requirements. There is no GUI at all; the entire user interface is implemented with character cell terminal. Rally script is not an object-oriented language. Code reuse is very difficult. Sophisticated algorithms and database processing need to be maintained externally in other environments. Thus Rally does not comply with the modern requirements of rich graphical user interface, n-tier architecture, component-oriented and service-oriented design.

And, the last but not least, Rally is an unsupported platform.

Nowadays more and more organizations are urged to move their Rally applications to a modern thin-client environment. There are 3 choices, which can be considered:

1. Face lifting
2. Manual rewriting from scratch
3. Automated conversion

Face lifting uses the existing application as a back-end, and therefore does not really move it to the modern supported environment.

Rewriting Rally applications from the scratch is a very costly, time consuming process and may lead to the long learning curve for the end users and to the loss of business knowledge. The significant point is that legacy applications are not always well documented. Even when an organization has the original system analysis document, not all the maintenance changes may have been documented so the risk of business logic loss is very high.

Conversion as a replacement methodology saves all the investments in business knowledge and opens the way to maintenance and further development in a new modern environment.

This document will present the conversion process suggested by MainTrend in order to meet all aspects and challenges of converting Rally applications to a full J2EE thin-client Java application.

MainTrend is today the sole provider of such a comprehensive solution for Oracle Rally applications.

Migration Methodology

Rally 4GL for rapid application development was adopted by many enterprises for their vertical application development in the late 1980's and early 1990's. In the context for which it was created, Rally was a formidable tool that allowed development of applications in the Open VMS environment for text-based systems. Many enterprises and organizations built their mission critical systems with Rally. Many man-years were invested in Rally code, in the application user interface and business logic. But now Rally applications have become legacy ones.

Currently enterprises are moving over to new technologies. Graphical User Interface (GUI) based operating systems, Internet, Mobile computing, e-commerce and other forces require businesses to move forward. Maintenance of old architectures is expensive and suppliers are discontinuing services and maintenance contracts. Professional personnel are hard to find to support old technologies. Users are demanding seamless integration of legacy systems with new technologies.

Moving to another platform requires a number of factors to be considered. One major consideration is the enterprise's investment in its legacy applications and significant cost of the new application development. Another consideration is how to ensure a rapid and smooth migration path from one infrastructure to another without disturbing critical systems. The more valuable the encapsulating an organization's business logic software is, the more complex is the process of its evolution. Maintaining legacy software business logic is the important part of an organization's knowledge storage. Smooth and reliable transformation requires preserving all of the needed business logic.

Rewriting Rally applications from the scratch is a very costly, time consuming process and may lead to the loss of business logic and user interface investments. And, the significant point is that legacy applications are not always well documented. Even when an organization has the original system analysis document, not all the maintenance changes may have been documented, so the risk of business logic loss is very high.

Automated conversion as a migration methodology saves all the investments in business logic and opens the way to maintenance and further development in a new modern environment.

The time to start migration is determined by the moment the new requirements can not be implemented with the old technology. The main goal is not only to migrate into a new environment, but to have an opportunity to implement new technologies. The right migration strategy also means changing the software ideology to meet new requirements. Therefore, just emulating the old legacy environment with the new one is a wrong decision.

Rally to Java (R2J) is a software engine toolset and set of methodologies that take an application coded in Rally and generate Java / Xml application code with minimal need for a programmer intervention. Java as a target environment allows maximum platform flexibility and at the same time allows, if needed, to remain with the current server-side platform. Java has proven itself to be a superior technology as a modern business application tool. Moving to Java is a strategic decision for an enterprise, a decision to make another step into the future. An enterprise that decides to migrate to this modern development environment will use Rally to Java (R2J) as its conversion tool.

Source Environment

Rally is a fourth generation language that unites a high level scripting language with extensive database access support, screen forms and reports. For its time Rally was a powerful, elegant and flexible application development tool. It did, however, have its limitations and does not meet modern requirements of rich graphical user interface, n-tier architecture, component-oriented and service-oriented design.

A Rally application consists of three major parts that should be addressed during the conversion process:

- DSD's ("data source definitions") – which specify a source of data that can be accessed from an application. The specification includes the names of the databases, tables, domains, and files to be used.
- ADL's ("application development language") – optional business logic procedures. For example, you may decide to define an ADL procedure to provide more sophisticated validation on user input.
- Menus and Forms – based on character-cell user interface.

Rally limitations:

- Code executed entirely on server
- Sophisticated algorithms and database processing cases need to be maintained externally in other environments
- Weak client server version abandoned
- No Graphical User Interface – character cell terminal implementation
- Not object oriented
- Not strong on code reuse
- Not "Open"
- No support for web delivery of forms
- No support for n-tier architecture
- Very few reported bugs
- Rally is no longer supported by Oracle
- New VMS versions and platforms do not support Rally

Target Environment

The target environment is a full thin client solution based on the modern J2EE and AJAX methodologies.

The new architecture is a pure thin client (“browser”) solution with a J2EE container and web server on the server side; the application is packed as a J2EE servlet running within a J2EE container; database access and some business logic are provided via J2EE components. All the data access layer (DAL) and presentation layer (GUI) object definition XML files are also stored on the central web server.

The server side can be any J2EE container / Java Application Server, running on any Java supporting operating system (IBM WebSphere, BEA WebLogic, Oracle Application Server, Tomcat, JBoss, Sybase EAServer, etc.).

The resulting application requires no client installation, except of a browser, with no operating system limitations. Currently supported browsers are Internet Explorer 6.0 and higher, and Firefox 3.0 and higher. The same browser software version has to be installed on all the client workstations. All the browser instances on the client workstations should have the same settings (security, JavaScript etc.).

Application changes, which are made to the XML definition files (database access definitions and GUI definitions), are available immediately. Business logic changes, encapsulated within application server components, are available immediately after installation. Business logic and GUI framework changes, encapsulated within the J2EE Servlet, are available immediately after the Servlet container restart. The database is accessed from the application server components.

This architecture provides centralized application management and does not require any additional software to be installed on the end-user workstations.

Migration Process

Generally the migration process has the following steps:

1. Detailed analysis, choosing of the target platform
2. Verification and fixing of the source application
3. Rally application report generation
4. Reverse engineering of the source application (based on the application report) and the database metadata
5. Manual completion of the unreported parts (aggregate fields' information, menu texts etc.)
6. Code generation
7. Preparing test environment
8. Database migration (if a new database platform is chosen)
9. Unit testing, including required database connection for the data access layer objects
10. External links migration
11. Code integration and thin platform adaptation. Application restructuring according to the thin client conversion model.
12. Database integration and required changes according to the chosen database platform
13. General graphical design, fine tuning of the resulting GUI in line with customer standards
14. Application integration testing and required fixing
15. User acceptance
16. Building the production environment and putting the new application to production
17. Trainings for the customer's developers

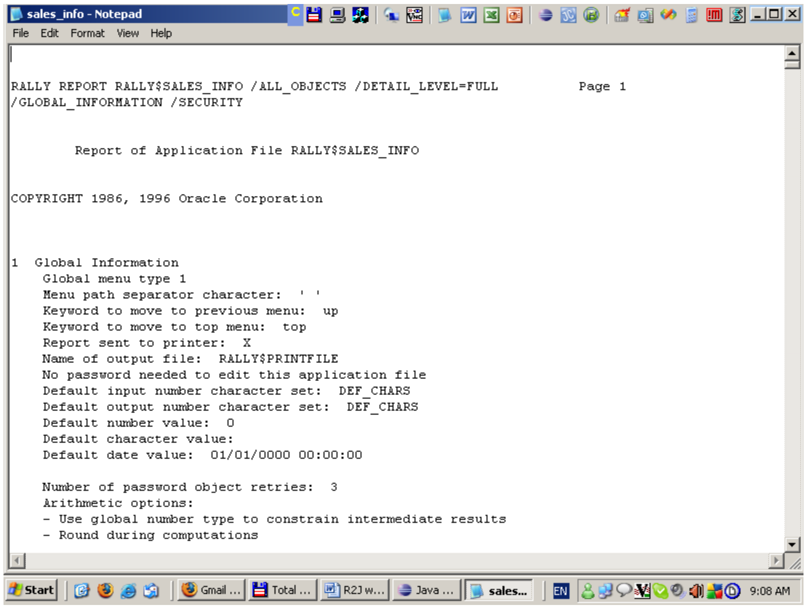
The conversion itself is covered by steps 3 through 6 and uses Rally Conversion Suite. Rally Conversion Suite consists of these parts:

- Java-based JDataPanel framework that allows future flexible and reliable development in the new Java-based environment, and which also includes an independent product for rich Form/Report design – MainTrend's JDataPanel Designer. The framework supplies all the needed functionality that the Rally application has, including database access, forms and reports. JDataPanel can be used to create new Java applications, not just to convert existing Rally applications.
- Rally Converter itself, which translates each Rally application into a set of Java and XML modules.
- Conversion and development methodologies, including best practices for the most effective manual completion and enhancement of the resulting applications.

The automated conversion process converts more than 80% of a Rally application to Java/XML automatically. That means that at least 80% of the total conversion work is performed automatically.

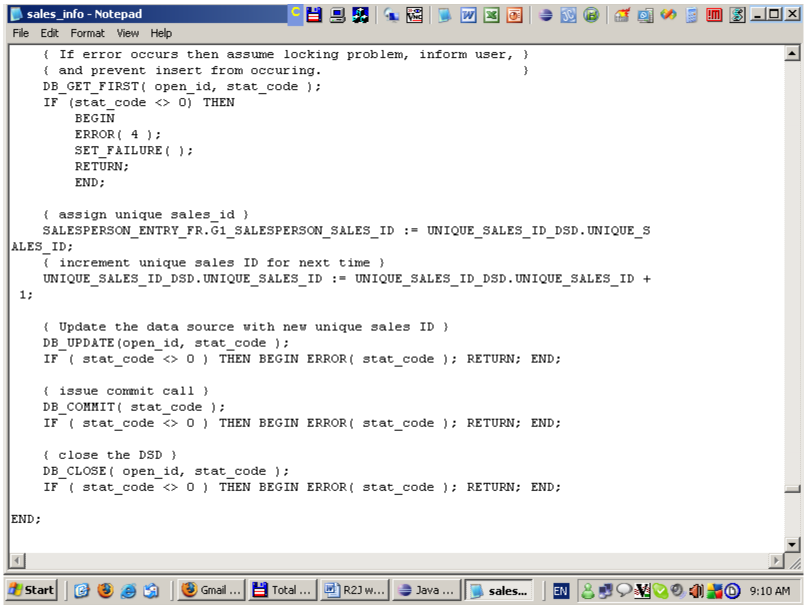
We use Rally application report as a source for our conversion process, and there is some information (such as types of aggregate fields, menu choice texts etc.) that is not included into the Rally report. This information can be easily added manually with our editing tool. Also there are some objects (external links, for example), which should be treated manually. For such objects we build wrapper classes, so the manual intrusion is encapsulated within those classes. Also manual part includes GUI tuning according to the customer's requirements and database access tuning. The database access is very significant, because Rally itself uses "connected" database access model and is tightly coupled with Rdb, and this approach is not always good for modern applications, and always not good for n-tier applications. From our experience, manual part can vary from 5% to 20%, depending on the application.

Here is an example of a Rally Application Report’s fragment:



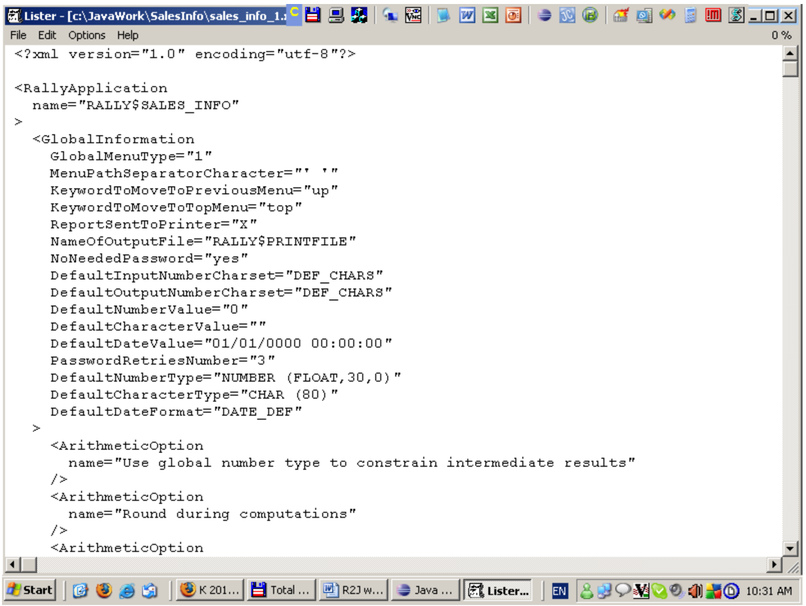
```

RALLY REPORT RALLY$SALES_INFO /ALL_OBJECTS /DETAIL_LEVEL=FULL          Page 1
/GLOBAL_INFORMATION /SECURITY


        Report of Application File RALLY$SALES_INFO


COPYRIGHT 1986, 1996 Oracle Corporation



1  Global Information
    Global menu type 1
    Menu path separator character:  ' '
    Keyword to move to previous menu:  up
    Keyword to move to top menu:  top
    Report sent to printer:  X
    Name of output file:  RALLY$PRINTFILE
    No password needed to edit this application file
    Default input number character set:  DEF_CHARS
    Default output number character set:  DEF_CHARS
    Default number value:  0
    Default character value:
    Default date value:  01/01/0000 00:00:00

    Number of password object retries:  3
    Arithmetic options:
    - Use global number type to constrain intermediate results
    - Round during computations
```

Another fragment, it can be noticed, that the text line length is limited by 80 characters, and therefore any word can be cut in unpredictable location:



```
    { If error occurs then assume locking problem, inform user, }
    { and prevent insert from occuring.                          }
    DB_GET_FIRST( open_id, stat_code );
    IF (stat_code <> 0) THEN
        BEGIN
        ERROR( 4 );
        SET_FAILURE( );
        RETURN;
        END;

    { assign unique sales_id }
    SALESPERSON_ENTRY_FR.G1_SALESPERSON_SALES_ID := UNIQUE_SALES_ID_DSD.UNIQUE_S
ALES_ID;
    { increment unique sales ID for next time }
    UNIQUE_SALES_ID_DSD.UNIQUE_SALES_ID := UNIQUE_SALES_ID_DSD.UNIQUE_SALES_ID +
 1;

    { Update the data source with new unique sales ID }
    DB_UPDATE(open_id, stat_code );
    IF ( stat_code <> 0 ) THEN BEGIN ERROR( stat_code ); RETURN; END;

    { issue commit call }
    DB_COMMIT( stat_code );
    IF ( stat_code <> 0 ) THEN BEGIN ERROR( stat_code ); RETURN; END;

    { close the DSD }
    DB_CLOSE( open_id, stat_code );
    IF ( stat_code <> 0 ) THEN BEGIN ERROR( stat_code ); RETURN; END;

END;
```

R2J framework contains sophisticated parser that takes this flat, free-text report, generated by the Rally application, into a multidimensional Object based XML file. The parser includes many lexical analysis heuristic algorithms for complex texts with free structure and unpredicted lexical hindrances.



```
<?xml version="1.0" encoding="utf-8"?>

<RallyApplication
  name="RALLY$SALES_INFO"
>
  <GlobalInformation
    GlobalMenuType="1"
    MenuPathSeparatorCharacter="' '"
    KeywordToMoveToPreviousMenu="up"
    KeywordToMoveToTopMenu="top"
    ReportSentToPrinter="X"
    NameOfOutputFile="RALLY$PRINTFILE"
    NoNeededPassword="yes"
    DefaultInputNumberCharset="DEF_CHARS"
    DefaultOutputNumberCharset="DEF_CHARS"
    DefaultNumberValue="0"
    DefaultCharacterValue=""
    DefaultDateValue="01/01/0000 00:00:00"
    PasswordRetriesNumber="3"
    DefaultNumberType="NUMBER (FLOAT,30,0)"
    DefaultCharacterType="CHAR (80)"
    DefaultDateFormat="DATE_DEF"
  >
    <ArithmeticOption
      name="Use global number type to constrain intermediate results"
    />
    <ArithmeticOption
      name="Round during computations"
    />
    <ArithmeticOption
```

The generated XML-representation information is compared with the Rally application (R2J suite automated tools are used for such comparison), and necessary fixes are performed manually.

Then this intermediate XML-representation is used as a source for the code generation. The generated objects are grouped into the following packages:

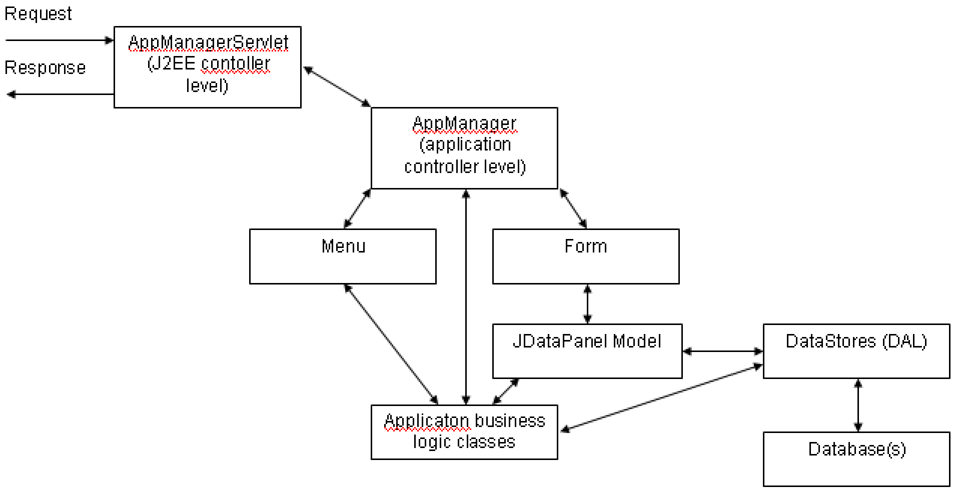
- “dal” – XML-definitions of the data access layer objects (DSDs)
- “external” – all the Java classes for external links
- “façade” – all the Java classes for Menus, Tasks and the application manager
- “logic” – all the Java classes for ADLs and Action Lists
- “presentation” – all the Java classes for Forms, Form Packets and all the XML-definitions for the GUI objects

The presentation layer object definitions and data access layer object definitions (XML files) can be stored with the resulting Java application, or can be deployed on a web server

The generated application is deployed with the “thin” framework as follows:

```
[webapps]
  [... the application's directory
    [css]
    [html]
    [images]
    [js]
    [jsp]
    [META-INF
      MANIFEST.MF
    ]
    [WEB-INF
      [classes
        [The application’s Java classes structured by package name:
          [dal]
          [external]
          [facade]
          [logic]
          [presentation]
        ]
        [net]
          [maintrend - MainTrend's engine Java classes]
        ]
      [lib
        [all the needed libraries, including Oracle JDBC driver]
      ]
      web.xml
    ]
    index.jsp
    login.jsp
    Configuration.xml
  ]
]
```

Logically the converted application works within a J2EE container as a servlet, which provides all the needed for the thin client environment functionality:


Request
Response
AppManagerServlet (J2EE contoller level)
AppManager (application controller level)
Menu
Form
JDataPanel Model
DataStores (DAL)
Applicaton business logic classes
Database(s)

Success Story

Family Department of the Commonwealth of Puerto Rico Government (“Departamento de la Familia”, DLF) is a large U.S. government agency, which administrates and is responsible of all forms of public assistance and social services offered to families in their needs and in the protection and care of children.

The Department has more then 3000 users in more than 100 local offices spread all over the Puerto Rico.

DLF had a legacy application, which supported almost all its activities, containing approximately 100,000 lines of code (estimated as more than 30 man-years of investment). This application was developed in Oracle Rally platform, which is no longer supported by Oracle, and is tightly coupled with Open VMS operating system and Oracle Rdb database.

DLF began experiencing many different problems that were attributed both directly and indirectly to this application:

- The technology was significantly outdated, leaving little or no room for the application to grow to the level of robustness required for the future.
- The existing Rally application required a complex emulator installation on each workstation.
- The lack of programmers’ availability and limitation of the platform prevented future developments and caused major issue on maintenance.
- The application lacked organized documentations that could lead to a potential business knowledge lost.

As a result of these problems, DLF decided that this technology was no longer feasible to maintain and began to seek out an alternative solution. Given the size and required reliability of the application, HP UNIX operating system was chosen as a target server platform, Oracle 10g database was chosen as a target database platform, and Java (J2EE) thin client solution was chosen as a target programming environment.

When the migration project was planned, the following considerations were taken into account in addition to mentioned problems:

- The project had short time frame and very tight budget.
- The number of users was significant, so the overall cost (both time and money) of assimilation would be very high.

As a solution MainTrend’s Rally to Java conversion service was chosen:

- Using MainTrend’s R2J Toolset to convert the application to a modern J2EE application.
- Using thin client (browser) as the client-side application container using MainTrend’s JDP framework.

A joint team (Hewlett Packard of Puerto Rico as a primary contractor, MainTrend as a subcontractor) undertook the entire migration project. It took less than 7 months from the project start to the new converted application working in production, and about one third of the estimated budget of the full rewriting.

All the goals of the migration project were achieved.

The project was completed and launched on September 2 2008. Since that time there were only two minor bug fixes. The converted system now supports all the 3000 users in 100 sites.

Business Results:

- The project took less then 7 month's, and was performed within the original budget and time frame.
- The resulting Java application is now maintained by the department's own developers.
- No loss in business knowledge.
- A significant reduction in client maintenance was achieved (Only browser maintenance is needed)
- Because there was no change in business logic and Forms structure, the assimilation efforts remain low.
- The new technology (Java rather than Rally) makes it much easier to find programming personnel.
- The more flexible programming environment allows a reduction in the "maintenance cost per change" factor.
- The new technology allows application changes which increase business productivity of the users

Summary

This White Paper describes the MainTrend's path for successful Oracle Rally migration into new advanced J2EE environment.

This migration methodology provides:

- Rapid, accurate and valid way for organizations to migrate Rally applications to Java
- Low assimilation costs for the new application
- The same productivity and more, in a new modern environment.
- No limitations for the target server-side platform: any Java-supporting platform will fit. A VMS/Rdb based organization can proceed without a major change in its structure.
- Much lower cost and faster solution for the legacy applications' migration path.
- Flexible open-platform development environment. The applications can be further developed by our GUI designing tools and framework alongside and integrated with the most cutting-edge commercial development tools
- Code reuse enabled environment and therefore reduced maintenance costs.
- Seamless integration with modern technologies and new approaches.

Reducing the conversion process significantly, R2J frees organizations from the translation process so they can continue to do what they do best — remain dedicated to running their core business functions instead of dedicating themselves to one usually lengthy migration project.

By working with the MainTrend's team of specialists, organizations receive the added benefit of applying our experience and advanced expertise to their unique and often complex requirements.

Appendix A:
JDataPanel (JDP) Framework

JDataPanel Framework is a central element of MainTrend’s programming environment. It is both a base for all the conversions (Rally-to-Java, PowerBuilder etc.), and also the independent framework for building J2EE applications. When developed, the purpose of the framework was not to write a Rally engine using Java as a programming language, rather to build a general framework that included Rally functionality and PowerBuilder’s DataWindow as special cases. The purpose was to give Java developers a powerful and flexible Java and XML based environment for building Data Access Layer and Presentation Layer components.

JDataPanel Framework consists of the following parts:

- JDataPanel components (Presentation layer)
- Data manager components (Data access layer)
- Application manager components (Control layer)
- Graphical designers (JDataPanel Builder and DataStore Builder)

JDataPanel is a universal XML-based user interface description:

- It has both "wide" and "deep" architectures: correlates with intuitive concept of object with attributes (“wide”) and sub-objects (“deep”).
- It has built-in integration with the Business Logic and Data Access Layers together with readiness for layers separation.
- It is a powerful form / report framework.

User Interface Description Structure

```
<?xml version="1.0" encoding="utf-8"?>
<group name="Main" ... (other attributes) >
  <textgroup name="t1" fontfamily="Courier New" x="0" ...>
    <text name="t1_1" container="t1" ...>Hello!</text>
  </textgroup>
  <group name="Header" datasource="DS1" ...>
    <text .../>
    <field name="date" datafield="indate" datatype="Date" ...>
      <editcontrol name="JDateField".../>
    </field>
    <group ...>
      <field .../>
      <text .../>
    </group>
  </group>
</group>
```

JDP framework supports an MVC (Model – View – Controller) design pattern even in thin client environment (Renderer + browser as “View”, JDP Model as “Model” and JDP commands as “Controller”).

DataStore part of the framework is used as a “Data Model” part of the “Model” in MVC pattern. DataStore framework is also a universal data access layer solution, and can be used independently from JDataPanel part of the JDP framework.

DataStore objects have their xml representation, which looks as follows:

```
<?xml version="1.0" encoding="utf-8"?>

<datastore name="EXAMPLE_DSD" clearOnRetrieve="true" comment=""
deleteAllowed="true" insertAllowed="true" retrieveLimit="0"
trimEndOnRetrieve="false" updateAllowed="true" updateFields="CODE,
DESCRIPTION" updateKey="CODE" updateSourceName="EXAMPLE_TBL"
definition="select EXAMPLE_TBL.CODE, EXAMPLE_TBL.DESCRIPTION from
EXAMPLE_TBL EXAMPLE _TBL order by EXAMPLE_TBL.CODE asc" adapter="1">

  <field name="CODE" dataType="String" sourceAlias="EXAMPLE_TBL"
sourceFieldName="CODE" sourceName="EXAMPLE_TBL" updateable="true"/>

  <field name="DESCRIPTION" dataType="String"
sourceAlias="EXAMPLE_TBL" sourceFieldName="DESCRIPTION"
sourceName="EXAMPLE_TBL" updateable="true"/>

</datastore>
```

Flexibility of the DataStore framework is based on the encapsulation of the data access functionality with the “adapters” concept. Each type of the data storage can be accessed with the appropriate adapter:

- SQLAdapter -> JDBC -> Database
- XMLAdapter -> XML files
- InMemoryAdapter (used to transfer the data between DataStores with the same application)

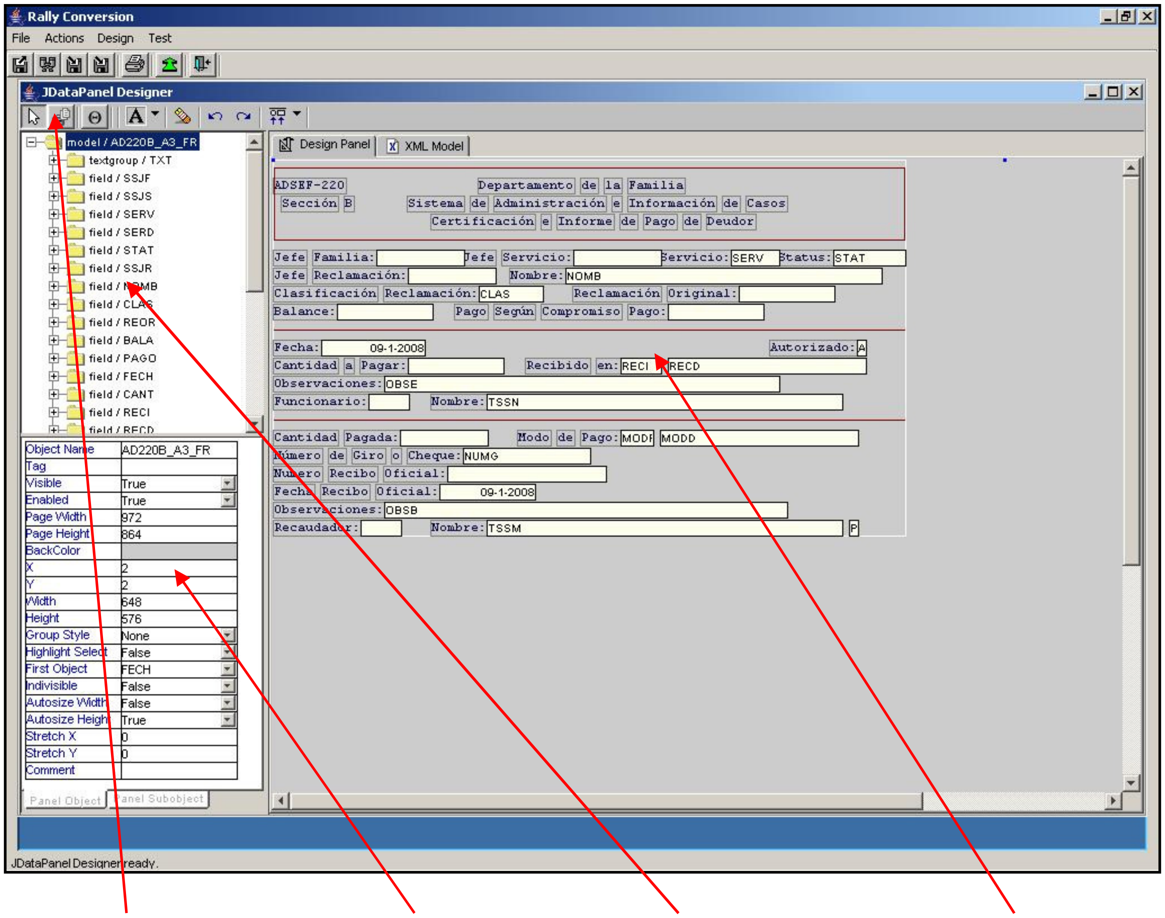
The adapters’ concept is open and any needed additional adapter can be easily developed.

Application manager components are the base for the building thin client solutions for converted from different sources applications, or for the applications which are built from scratch with JDP framework.

These components can be easily inherited and extended to suite all the needs of the chosen target platform.

To increase the development productivity and to help the developers in working with JDP framework, the Graphical designers (JDataPanel Builder and DataStore Builder) can be used:

Here is the example of the JDataPanel (form), opened within JDataPanel Builder:



Object Name	AD220B_A3_FR
Tag	
Visible	True
Enabled	True
Page Width	972
Page Height	864
BackColor	
X	2
Y	2
Width	648
Height	576
Group Style	None
Highlight Select	False
First Object	FECH
Indivisible	False
Autosize Width	False
Autosize Height	True
Stretch X	0
Stretch Y	0
Comment	

ADSEF-220 Departamento de la Familia
Sección B Sistema de Administración e Información de Casos
Certificación e Informe de Pago de Deudor

Jefe Familia: Jefe Servicio: Servicio: SERV Status: STAT
Jefe Reclamación: Nombre: NOMB
Clasificación Reclamación: CLAS Reclamación Original:
Balance: Pago Según Compromiso Pago:

Fecha: 09-1-2008 Autorizado: A
Cantidad a Pagar: Recibido en: RECI RECD
Observaciones: OBSE
Funcionario: Nombre: TSSN

Cantidad Pagada: Modo de Pago: MODI MODD
Número de Giro o Cheque: NUMG
Numero Recibo Oficial:
Fecha Recibo Oficial: 09-1-2008
Observaciones: OBSB
Recaudador: Nombre: TSSM P

JDataPanel Designer ready.

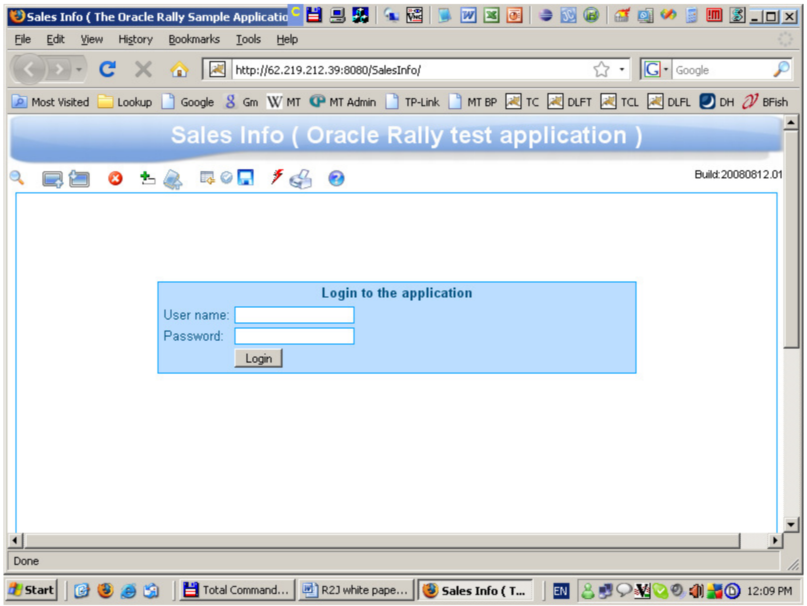
Toolbar Area Properties Panel Object Hierarchy Preview Area

Appendix B:

Out-of-the-box Look and Feel of the Converted Application

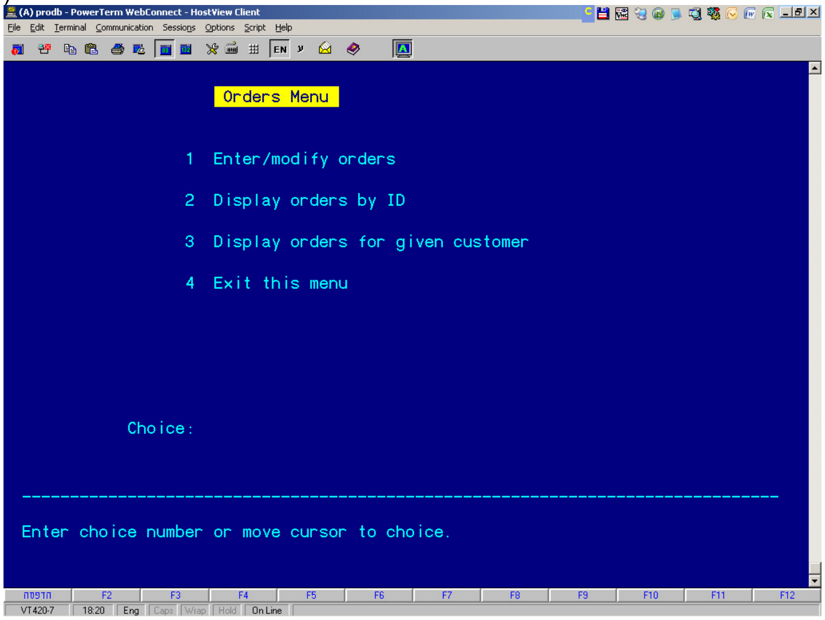
“Sales Info” is a test application, which is supplied with Oracle Rally programming environment. The following screenshots are intended to give the feeling of look and feel of the converted application without any graphical tuning. The application can be also accessed at http://62.219.212.39:8080/SalesInfo/

The first page (login page) looks like the following:


Sales Info (The Oracle Rally Sample Applicatio
File Edit View History Bookmarks Tools Help
http://62.219.212.39:8080/SalesInfo/
Google
Most Visited Lookup Google Gm MT MT Admin TP-Link MT BP TC DLFT TCL DLFL DH BFish
Sales Info (Oracle Rally test application)
Build:2008081201
Login to the application
User name:
Password:
Login
Done
Start
Total Command...
R2J white pape...
Sales Info (T...
EN
12:09 PM

Here is the Orders Menu:

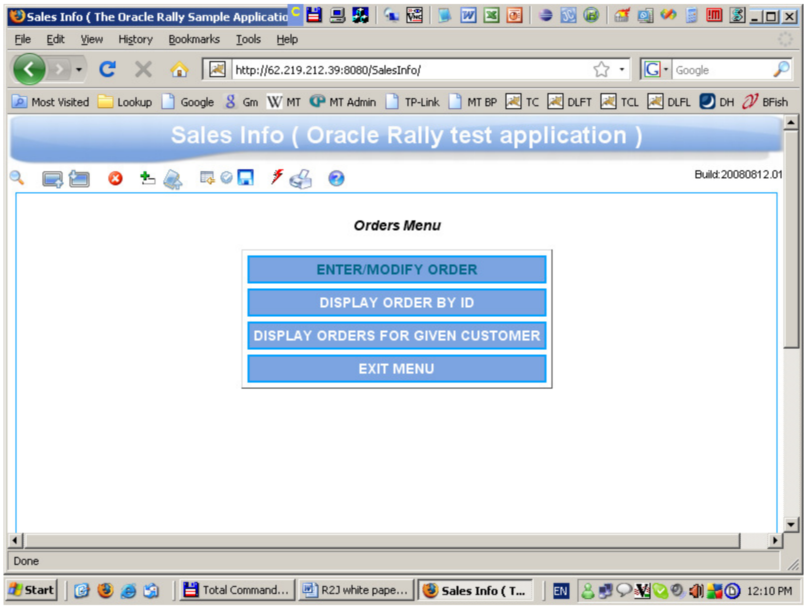
Original (Rally):



```
                    Orders Menu


                 1  Enter/modify orders

                 2  Display orders by ID

                 3  Display orders for given customer

                 4  Exit this menu




         Choice:


-------------------------------------------------------------------------------

Enter choice number or move cursor to choice.
```

(A) prodb - PowerTerm WebConnect - HostView Client
File Edit Terminal Communication Sessions Options Script Help
F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 18:20 Eng Caps Wrap Hold On Line

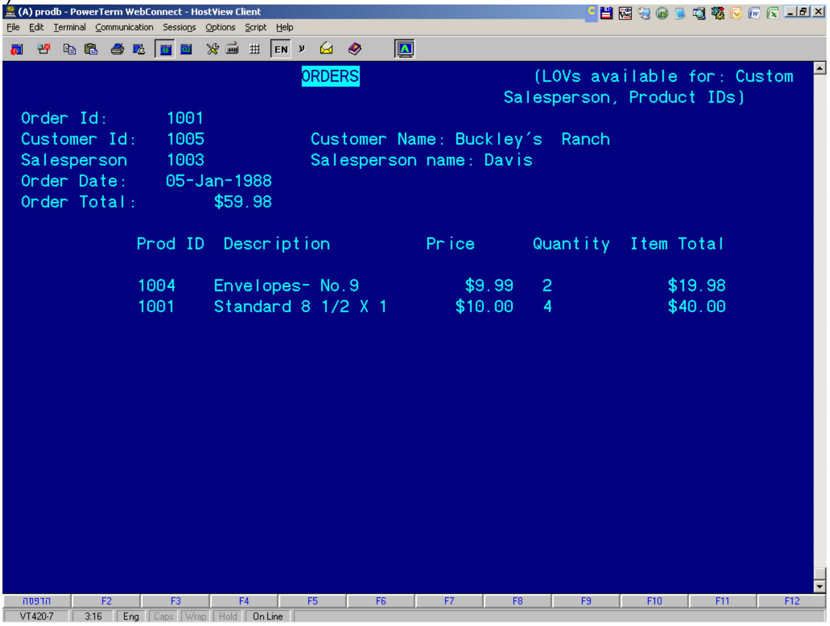
Converted:



Sales Info (The Oracle Rally Sample Applicatio
File Edit View History Bookmarks Tools Help
http://62.219.212.39:8080/SalesInfo/
Google
Most Visited Lookup Google Gm MT MT Admin TP-Link MT BP TC DLFT TCL DLFL DH BFish
Sales Info (Oracle Rally test application)
Build: 20080812.01
Orders Menu
ENTER/MODIFY ORDER
DISPLAY ORDER BY ID
DISPLAY ORDERS FOR GIVEN CUSTOMER
EXIT MENU
Done
Start
Total Command...
R2J white pape...
Sales Info (T...
EN
12:10 PM

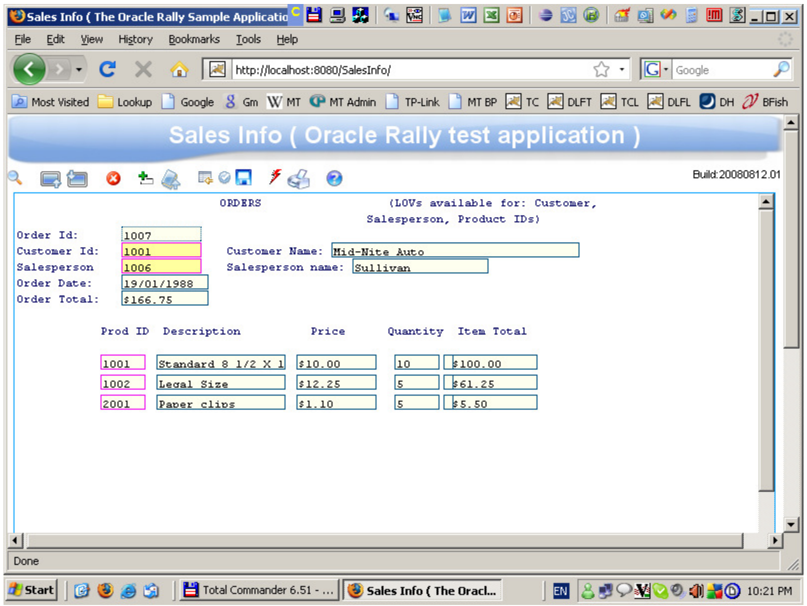
And here is Orders form:

Original (Rally):


(A) prodb - PowerTerm WebConnect - HostView Client
File Edit Terminal Communication Sessions Options Script Help
EN
F2 F3 F4 F5 F6 F7 F8 F9 F10 F11 F12
VT420-7 3:16 Eng Caps Wrap Hold On Line

```
                              ORDERS                 (LOVs available for: Custom
                                                  Salesperson, Product IDs)
 Order Id:      1001
 Customer Id:   1005               Customer Name: Buckley's  Ranch
 Salesperson    1003               Salesperson name: Davis
 Order Date:    05-Jan-1988
 Order Total:        $59.98

             Prod ID  Description          Price      Quantity  Item Total

             1004     Envelopes- No.9         $9.99    2             $19.98
             1001     Standard 8 1/2 X 1     $10.00    4             $40.00
```

Converted:


Sales Info (The Oracle Rally Sample Applicatio
File Edit View History Bookmarks Tools Help
http://localhost:8080/SalesInfo/
Google
Most Visited Lookup Google Gm MT MT Admin TP-Link MT BP TC DLFT TCL DLFL DH BFish
Sales Info (Oracle Rally test application)
Build:20080812.01
Done
Start
Total Commander 6.51 - ...
Sales Info (The Oracl...
EN
10:21 PM

```
                         ORDERS                 (LOVs available for: Customer,
                                             Salesperson, Product IDs)
Order Id:         1007
Customer Id:      1001          Customer Name:  Mid-Nite Auto
Salesperson       1006          Salesperson name:  Sullivan
Order Date:       19/01/1988
Order Total:      $166.75

           Prod ID  Description           Price        Quantity  Item Total

           1001     Standard 8 1/2 X 1    $10.00       10        $100.00
           1002     Legal Size            $12.25       5         $61.25
           2001     Paper clips           $1.10        5         $5.50
```

The conversion services, pricing estimation and model

We offer an automated conversion service that can lift Rally applications to the Java/XML technologies while preserving business logic base and user interface investments.

We do not sell the tool itself, we offer a service based on our tool. Most of the work can be done remotely, and manual part can be divided between MainTrend and a customer or a third party conversion partner, to be as convenient as possible to all the parties.

There can be different models of the conversion projects - from full conversion to different "reverse engineering" stages, automatically generated script can be used by customer's developers as a tool for business logic capturing, etc. Business model can also vary - from fixed price projects to hourly based consulting. All the mixed type models are also available.

As a first step we can define a pilot for the conversion project. The pilot will give enough information to prove that the conversion suits the application; and the pilot results can be used then as a part of the whole conversion.

To be more precise in our estimations we need the answers to the following questions:

- What is the total size of your Rally applications (in VMS blocks)?
- Is your business logic mostly incorporated within Rally application itself, or located externally? What are your plans for the converting of the external business logic routines?
- What will be your server-side environment (operation system, database, application server etc.)?
- What will be your network topology?
- What is your estimation for the overall quantity of end-users?
- Are your personnel familiar with Java environment and, if so, on what level?
- What is your schedule for the project?

Based on this information we can make (though very preliminary) estimation of the needed efforts and time schedule, and build the right price model. More accurate estimation requires Rally applications and database structure to be analyzed.

In general for the pricing model we can suggest "mixed" model, that is fixed price based on a Rally application characteristics (number of VMS blocks, forms, fields, DSD's, external links, tasks etc.) plus consulting hours pool. From our experience, additional consulting is no more than 50% of the application-dependent price.

About MainTrend

Founded in 2002, MainTrend is a company that unites high-level information technology and organization architecture professionals. MainTrend is a software and solution provider, focused on integrative decisions for its customers.

Our professionals work in government structures, banks, private corporations and software development organizations; we know the needs of these types of organizations and are able to both supply integrative solutions and assist in their decision making processes.

Our professionals have extensive expertise in system architecture design, databases, application programming and decision making systems.

We are proud of our ability to perform complex projects, meeting planed budget, technology and time goals.