



Solution Technology Overview

TECHNICAL ARCHITETURE [TECHNOLOGY COMPONENTS]

Technology Component	Technology Used by T3
T3 Front end Technology	<p>T3 uses Oracle JDeveloper/ ADF Application Development Framework (Version 12.2.1.0.0) as a Frontend Technology for application development.</p> <p>JDeveloper covers the full development lifecycle from design through coding, debugging, optimization and profiling to deploying.</p> <p>With JDeveloper, Oracle has aimed to simplify application development by focusing on providing a visual and declarative approach to application development in addition to building an advanced coding-environment. Oracle JDeveloper integrates with the Oracle Application Development Framework (Oracle ADF) - an end-to-end Java EE-based framework that further simplifies application development</p>
T3 Middleware Technology	<p>T3 solution is build around 3-Tier Architecture.</p> <p>The Middleware Technology used in T3 solution is Glassfish Server Open Source Edition (Version 4.1) .</p> <p>GlassFish Server 4.1 supports Java EE seven, which provides the basis for delivering dynamic and scalable hypertext language applications. Key features include applied science Java seven compatibility, modularity, and rapid repetitive development.</p> <p>Note: This can also be deployed using Oracle Weblogic Server</p>
T3 Backend Technology	<p>T3 uses Amazon RDS for Oracle as a backend Technology.</p> <p>A database server is the key to information management. In general, a server reliably manages a large amount of data in a multiuser environment so that users can concurrently access the same data. A database server also prevents unauthorized access and provides efficient solutions for failure recovery.</p> <p>Oracle® Database is a relational database management system developed by Oracle. Amazon RDS makes it easy to set up, operate, and scale Oracle Database deployments in the cloud. AWS manages database administration tasks including provisioning, backups, software patching, monitoring, and hardware scaling.</p>
T3 Mobile Technology	<p>T3 Provides Android and iOS Based Mobile App. Employee and Manager Self Service’s applications are fully mobile responsive and designed using masonry layouts. These work using internet browser seamlessly like a dedicated mobile app and responsive to any device or size of screen.</p>
T3 Browser Support	<p>T3 applications can be accessed via Laptop, Mobile, Tablets using all commonly used web browsers like - Google Chrome, Firefox, Apple Safari, Microsoft Internet explorer, Microsoft Edge</p>
T3 Operating System Support	<p>T3 application can be deployed on both Linux and Windows operating environment.</p>

Oracle JDeveloper/ ADF

JDeveloper is a freeware IDE supplied by Oracle Corporation. It offers features for development in Java, XML, SQL and PL/SQL, HTML, JavaScript, BPEL and PHP. JDeveloper covers the full development lifecycle from design through coding, debugging, optimization and profiling to deploying.

With JDeveloper, Oracle has aimed to simplify application development by focusing on providing a visual and declarative approach to application development in addition to building an advanced coding-environment. Oracle JDeveloper integrates with the Oracle Application Development Framework (Oracle ADF) - an end-to-end Java EE-based framework that further simplifies application development.

The core IDE exposes an API that other teams in Oracle use to build extensions to JDeveloper. BPEL, Portal, Business Intelligence and other components of the Oracle platform[which?] all build their design-time tools on top of JDeveloper. The same IDE platform also serves as the basis of another Oracle product, SQL Developer, which Oracle Corporation promotes specifically to PL/SQL- and database-developers.

Key features of Oracle JDeveloper/ADF application development framework are as follows:-

- Platform and Database Independent
- Multi Faces & Device Independence
- Rich look and Feel, Smart Dashboards & Search tools
- Customizable, Easy to use, Smooth user Interfaces
- Component Reusability
- Mobile Application Development
- Take care of the "plumbing"
- Metadata driven – not code
- Technology chosen for Oracle's future business applications
- Drag-and-drop features
- Tight integration with Web Services inform of Web Service data control
- Declarative end-to-end security

Framework Richness in Functionality

One reason to take a close look at ADF is what it can do – it is that simple. The rich UI and powerful functionality that can be realized more or less out of the box is extremely attractive to any developer or development team.



Some screenshots of user interfaces created using ADF 11g

ADF comes with a large number of (JSF) components for presenting and editing data, laying out page and interacting through drag & drop, popups, server push and partial refresh. The integrated visualization components make it extremely easy to render data in a large number of interactive (drill down, popup) graphs. Complex navigation, simple reuse of both simple components and composite units (multi page, business logic), deep link navigation and personalization and customization are some of the out of the box functions available in ADF.

Powerful Data Visualization Tools

The ADF data visualization components provide significant graphical and tabular capabilities for displaying and analyzing data. These components provide the following common features:

- They are full ADF Faces components that support the use of ADF data controls.
- They provide for declarative design time creation using the Data Controls Panel, the JSF visual editor, Property Inspector, and Component Palette.
- Each component offers live data preview during design. This feature is especially useful to let you see the effect of your design as it progresses without having to compile and run a page.

Data visualization components include: graph, gauge, pivot table, geographic map, thematic map, Gantt charts, and hierarchy viewer.

The prefix `dvt:` occurs at the beginning of each gauge component name indicating that the component belongs to the ADF Data Visualization Tools (DVT) tag library.

Graph Component Use Cases and Examples

The graph component gives you the capability of producing more than 50 types of graphs, including a variety of bar graphs, pie graphs, line graphs, scatter graphs, and stock graphs. This component lets you evaluate multiple data

points on multiple axes in many ways. For example, a number of graphs assist you in the comparison of results from one group against the results from another group.

The following kinds of graphs can be produced by the graph component:

- **Area graph (areaGraph):** Creates a graph in which data is represented as a filled-in area. Use area graphs to show trends over time, such as sales for the last 12 months. Area graphs require at least two groups of data along an axis. The axis is often labeled with increments of time such as months.
- **Bar graph (barGraph):** Creates a graph in which data is represented as a series of vertical bars. Use to examine trends over time or to compare items at the same time, such as sales for different product divisions in several regions.
- **Bar (horizontal) graph (horizontalBarGraph):** Creates a graph that displays bars horizontally along the Y-axis. Use to provide an orientation that allows you to show trends or compare values.
- **Bubble graph (bubbleGraph):** Creates a graph in which data is represented by the location and size of round data markers (bubbles). Use to show correlations among three types of values, especially when you have a number of data items and you want to see the general relationships. For example, use a bubble graph to plot salaries (X-axis), years of experience (Y-axis), and productivity (size of bubble) for your work force. Such a graph allows you to examine productivity relative to salary and experience.
- **Combination graph (comboGraph):** Creates a graph that uses different types of data markers (bars, lines, or areas) to display different kinds of data items. Use to compare bars and lines, bars and areas, lines and areas, or all three.
- **Funnel graph (funnelGraph):** Creates a graph that is a visual representation of data related to steps in a process. The steps appear as vertical slices across a horizontal cylinder. As the actual value for a given step or slice approaches the quota for that slice, the slice fills. Typically a funnel graph requires actual values and target values against a stage value, which might be time. For example, use this component to watch a process (such as a sales pipeline) move towards a target across the stage of the quarters of a fiscal year.
- **Line graph (lineGraph):** Creates a graph in which data is represented as a line, as a series of data points, or as data points that are connected by a line. Line graphs require data for at least two points for each member in a group. For example, a line graph over months requires at least two months. Typically a line of a specific color is associated with each group of data such as Americas, Europe, and Asia. Use to compare items over the same time.
- **Pareto graph (paretoGraph):** Creates a graph in which data is represented by bars and a percentage line that indicates the cumulative percentage of bars. Each set of bars identifies different sources of defects, such as the cause of a traffic accident. The bars are arranged by value, from the largest number to the lowest number of incidents. A Pareto graph is always a dual-Y graph in which the first Y-axis corresponds to values that the bars represent and the second Y-axis runs from 0 to 100% and corresponds to the cumulative percentage values. Use the Pareto graph to identify and compare the sources of defects.
- **Pie graph (pieGraph):** Creates a graph in which one group of data is represented as sections of a circle causing the circle to look like a sliced pie. Use to show the relationship of parts to a whole such as how much revenue comes from each product line.
- **Radar graph (radarGraph):** Creates a graph that appears as a circular line graph. Use to show patterns that occur in cycles, such as monthly sales for the last three years.
- **Scatter/polar graph (scatterGraph):** Creates a graph in which data is represented by the location of data markers. Use to show correlation between two different kinds of data values such as sales and costs for top

products. Scatter graphs are especially useful when you want to see general relationships among a number of items.

- Sparkchart (`sparkChart`): Creates a simple, condensed graph that displays trends or variations, often in the column of a table or inline with text. Sparkcharts are simple in design, with limited features and formatting options, showing as much data as possible.
- Stock graph (`stockGraph`): Creates a graph in which data shows the high, low, and closing prices of a stock. Each stock marker displays three separate values.