

**IFS APPLICATIONS™ 7**  
**ARCHITECTURE AND TECHNOLOGY**

**WHITE  
PAPER**

Executive summary .....	3
Building for change .....	4
IFS Foundation1™ .....	5
IFS Service-Oriented Component Architecture .....	6
Components for agility and reliability .....	6
Service-oriented architecture with a twist .....	7
Multi-tier architecture .....	8
Integration through access providers and IFS Connect .....	9
Deployment view of the architecture .....	9
Windows, UNIX and Linux platforms .....	10
User interfaces .....	11
Optimizing the User Interface .....	12
IFS Web Client and portal .....	12
IFS Windows Client .....	14
IFS Mobile Client .....	14
Integrated learning .....	15
Integration, web services .....	16
Presentation tier integration—IFS Access Providers .....	16
Integrating processes and data flows (EDI, web services) .....	17
IFS Connect .....	18
Interface browser .....	19
Reporting and analysis .....	20
Transforming data into information .....	21
Operational reports—IFS Report Designer .....	21
Security .....	22
Secure by design and secure by default .....	23
Authorization and segregation of duties .....	23
Open and flexible network security .....	24
Tools .....	25
IFS Solution Manger .....	25
Custom Development .....	27
Sizing and scalability .....	28
IFS Applications sizing guides .....	29

# IFS APPLICATIONS™ 7 ARCHITECTURE AND TECHNOLOGY

This document provides a high-level overview of the technology and architecture used by IFS Applications 7. It is written for an audience familiar with software, Internet, and business application architecture concepts.

## Executive summary

Technology can be used to achieve competitive advantage, increased customer service and loyalty, and increased efficiency and profitability. But technology can also create pitfalls such as vendor lock-in or escalating costs. IFS develops the technology and architecture in IFS Applications for agility, ease of use, low total cost of ownership, and interoperability with other applications and technologies.

IFS Applications is built using the same principles employed by successful manufacturers—the production and assembly of components. This allows change and evolution within a component without affecting other components. With a proven track record of building for change, IFS provides step-by-step, nondisruptive introduction of new technologies into operational systems. This ensures that a company's business will always benefit from the latest advancements.

IFS' openness and commitment to standards protects investments from vendor lock-in. IFS Applications supports IBM®, Oracle®, and Microsoft® technologies alongside Linux, JBoss™ and other open source technologies to enable companies to use market competition to their best advantage.

In IFS Service-Oriented Component Architecture™ the agility and ease of use of service-oriented architectures (SOA) and the proven benefits of components and object orientation (OO) are combined to create a powerful architecture. Integration with other applications is supported through XML, web services, Java, J2EE and .NET technologies. IFS' web, portal, Windows®, and mobile user interfaces are designed to suit the particular roles, environments, and tasks that users have at hand.

Since business applications are mission-critical to the companies that use them, IFS pays special attention to product qualities such as security, scalability, and availability. Security is designed into IFS Applications, and the more secure configuration options are default. All server-side tiers, including database and application servers, can be clustered for maximum availability.

## Building for change

The world of business is constantly changing. So is the world of technology and computer software. IFS' philosophy is very simple—the key to success is to successfully manage change.

Change is everywhere. There is change in IT environments. New applications are being brought in, integrations created, hardware replaced, and some solutions phased out. There is change in software technology, with new paradigms, tools, and development languages. And there is change in application architecture.

IFS sees an investment in IFS Applications as an investment in an important business asset—not that different from a paper mill, a power plant, or an aircraft.

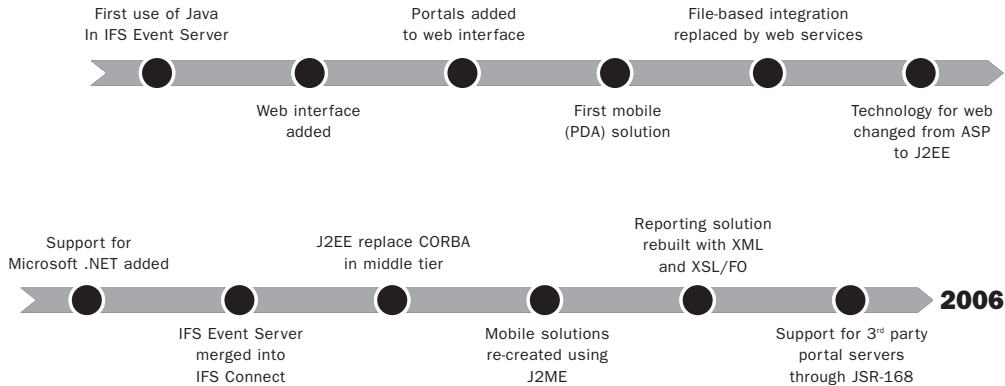
As with any asset, IFS Applications should be state-of-the-art when delivered, but also be cost-effective and able to generate returns throughout its lifecycle. Toward this goal, components must be continuously upgraded, added, and replaced with minimal downtime and impact on operations. To manage this, IFS has developed a strategy of building for change.

Building for change is a design and engineering challenge. Over the years, IFS has identified a number of best practices to meet this challenge:

- IFS invests time in, and carefully plans, the evolution of its architecture to build on existing investments and create a robust design that can accommodate future change.
- IFS takes the time to encapsulate and isolate architecture layers and platform components that might subsequently become subject to change.
- IFS supports both leading brand and open-source products so that companies are free to change the environment in which IFS Applications runs.
- IFS embraces commodities. Rather than continue to develop its own proprietary functionality, IFS uses third-party standard solutions whenever possible. This reduces legacy and prevents the company from getting stuck with overwhelming maintenance.

Most importantly, however, IFS' commitment to standard technologies and well-known products has many advantages. For instance, it separates concerns. By introducing standards between architecture tiers and platform components, IFS can replace and update technologies in isolated areas without causing ripple effects across the applications. Using standards also allows existing and widely available competence to be used, cutting start-up time and training costs for system integrators, developers, and technicians. Finally, adhering to standards and avoiding proprietary extension protects against vendor lock-in.

IFS' strategy of building for change applies also to standards. As new standards appear, IFS updates its architecture accordingly. XML, J2EE™, .Net, and web services were little known five years ago. Today they are all industry standards and integral parts of IFS Applications.



IFS has a long history of managing change and evolution in technology and architecture without disruption to core applications

## IFS Foundation1™

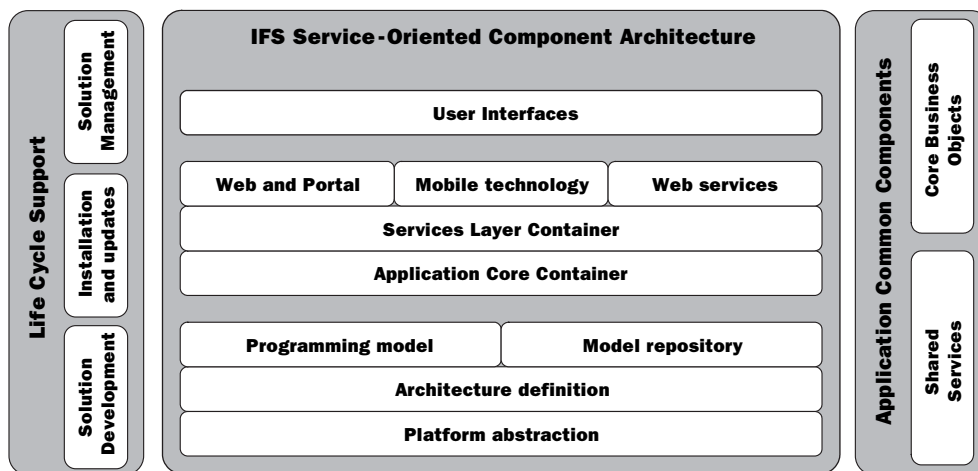
IFS Foundation1 is IFS' strategic packaged architecture platform for component-based business applications. Based on open standards such as XML, UML, BPEL, J2EE and .NET, it includes the technology and tools to design, develop, deploy, configure, integrate, and administer IFS Applications.

IFS Foundation1 is also the product family name for all things related to IFS technology and architecture. IFS Foundation1 includes:

- The definition and documentation of IFS Service-Oriented Component Architecture.
- Runtime containers, frameworks required to run and manage an IFS Applications installation. This includes runtimes for business logic, portals, web, Windows, and mobile clients.
- User interface environment enabling access to IFS Applications from web browsers, PCs, PDAs and more.

- Tools and routines to manage the full lifecycle of the applications. From solution development, through installation and configuration, to administration and use of IFS Applications.
- Options that extend the configuration possibilities of the platform. For example, IFS Connect and IFS Report Designer.
- Common application components and services such as ISO codes, security services, change logging, event notifications and user collaboration.

Everything in IFS Foundation1—from the choice of core technologies to the functionality in the tools—is optimized for the development of large, scalable, high-quality business applications.



IFS Foundation1 contains the definition and technology of IFS Service-Oriented Component Architecture, tools to support the full lifecycle, and common application components.

## IFS Service-Oriented Component Architecture

### Components for agility and reliability

IFS Applications is built using the principles employed by successful manufacturers—the production and assembly of components. Application solutions are built from 100+ business components, which in turn are built from 6000+ smaller software components that implement the functional aspects of the application. Each

component has a well-defined purpose and interface that describes what the component does, but not how it does it. This allows change and evolution within a component without affecting other components.

At the lowest level, components carry out basic functions such as retrieving or updating information in the database. Through successive layers, the business components come to represent complete business entities and activities such as customers and order management. Customers realize the benefits of this approach by choosing only the business components they need. Later, other components may be added or upgraded without disrupting the components already in place. This results in a very stable operational system in spite of change. A company's system can evolve in an orderly way without the "big-bang" approach required by a non-component-based, monolithic system.

IFS uses a combination of process modeling and Unified Modeling Language (UML) object modeling to design IFS Applications. The process model describes what the application does; the object model describes the components that make up the application and how they relate to one another.

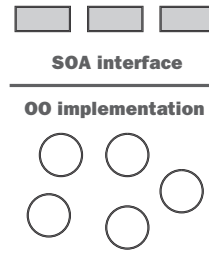
### Service-oriented architecture with a twist

IFS Service-Oriented Component Architecture combines the proven benefits of components and object orientation with the agility and ease of use of service-oriented architectures (SOA).

Object orientation has taken software development forward in the areas of productivity, reliability, and maintainability as a result of the extensive use of interfaces and the tight coupling between information and the functionality that operates on it. However, object orientation has failed to create a straightforward mapping of real-life business processes to software functions. To adapt software to changes and integrations in real-life processes, designers must find the relevant objects, understand their relationships, and figure out which parts of the objects apply to the task at hand. A large amount of detailed knowledge is required to effect change. This creates a time and cost barrier to change.

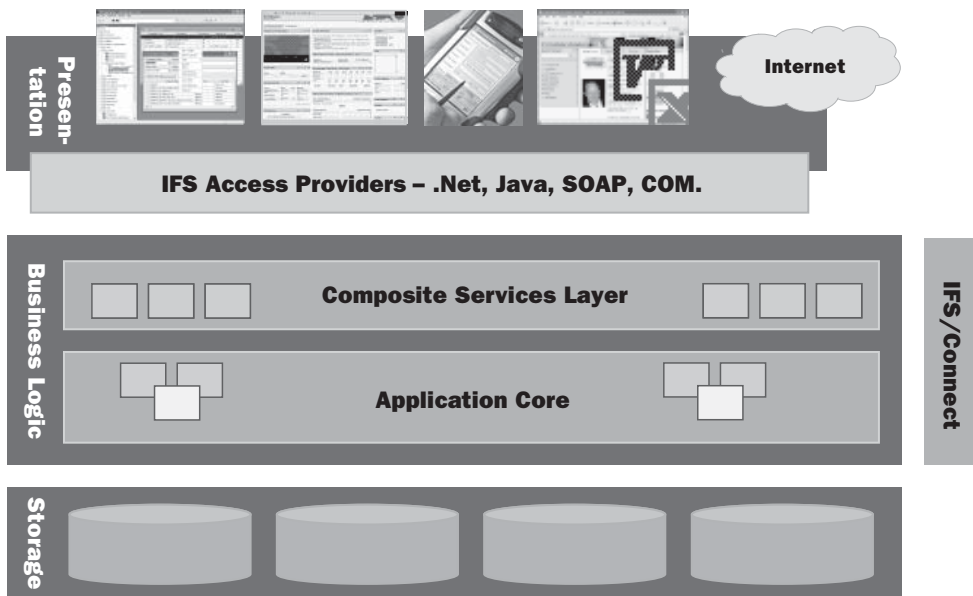
Process-oriented enterprise computing relies on SOA to address many of the issues in adapting software to change in real-life business processes. SOA applications map well to models and language used to describe real-life processes. In practice, such models as the Business Process Modeling Language (BPML) require the application to expose service interfaces or to be wrapped behind them. Where SOA promotes software functionality to be organized around the business service it provides, rather than which object it operates on, it also becomes easier to add and change process-related business rules in an SOA application.

Although SOA addresses many challenges related to developing agile business software, it suffers from some of the drawbacks that were solved by object orientation. For example, SOA applications generally have more redundant implementations, resulting in quality risks and more difficult maintenance. The many layers of indirection and relatively low level of maturity of development tools mean that performance can suffer—both when using the systems and during development. IFS has chosen to combine the strengths of both object orientation and SOA in IFS Service-Oriented Component Architecture. An object-oriented core provides the performance, reliability, and quality required in business applications. Combined with a service layer and hundreds of ready-to-use SOA services this makes it easy to integrate IFS Applications with other business systems, customers, and partners.



### Multitier architecture

IFS Service-Oriented Component Architecture is a layered, multitier architecture. Each tier has its own job to do and does it in a standard way with a clear separation from, and interface to, other tiers. Each tier has its own software objects. These are all derived from a common design model and implemented with technologies to suit their purpose.



IFS Service-Oriented Component Architecture is divided in to three main tiers, with the business logic available through access providers to IFS user interfaces and custom interfaces.

The fully normalized data storage tier is based on the relational database model. This is the prevalent storage technology for today and the foreseeable future. The database server is configured so that no data can be accessed or modified except through the business logic. This guarantees data integrity and prevents “back-door” modification.

The business logic tier is the heart of the application. It implements business knowledge, functionality, and processes. This tier is divided into two subtiers. The application core subtier is a high-performance, object-oriented implementation of business-object level and activity-level business logic. The composite services layer is a service-oriented interface and API used for integrations, client access, and process-level logic.

The presentation tier provides interaction with human users and client-side applications and devices. The same business logic can serve many different types of interfaces, such as web browsers, PDAs, and PCs. Interfaces in the presentation tier access the functionality of the business logic tier through access providers.

### Integration through access providers and IFS Connect™

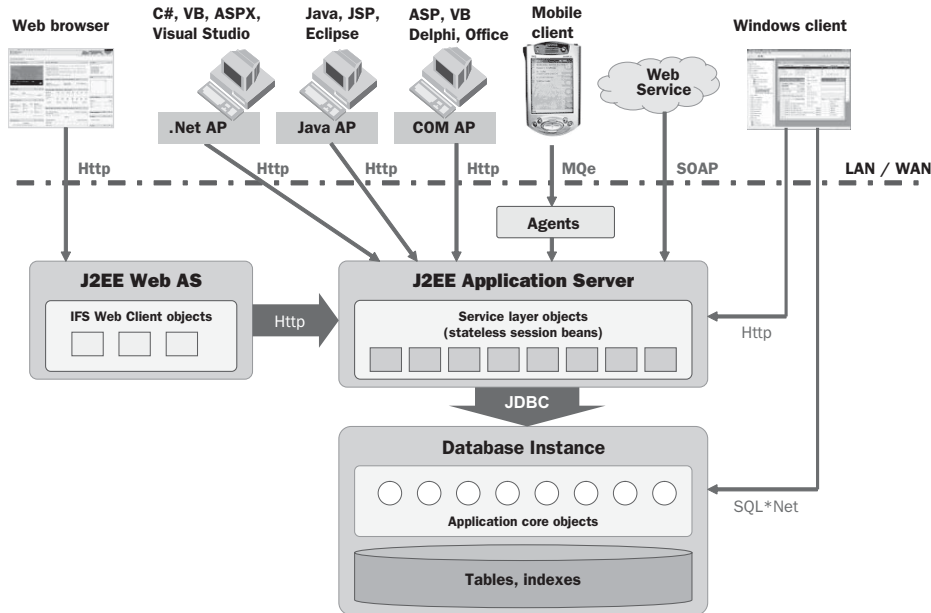
IFS Service-Oriented Component Architecture is an open architecture, allowing other applications to access information and invoke functionality in IFS Applications. Both the services layer and application core are exposed to other applications and environments through IFS’ access providers for .Net, COM, Java™, and SOAP. Message-based business process integration is provided through IFS Web Services and the IFS Connect integration broker.

### Deployment view of the architecture

IFS Applications is built using standard tools and technologies. This gives low total cost of ownership (TCO) and a high degree of freedom in deployment options. IFS supports most platforms, including Microsoft Windows, major UNIX® variants, and Linux®. Because each physical tier in the architecture is separated through standard protocols, it is possible to “mix and match” platforms in a deployment. IFS Applications can be deployed on anything from a single laptop running all components to multi-server installations with both database and application servers running in cluster configurations—supporting tens of thousands of users.

The two principal deployment components are the database and the J2EE application server. The open architecture makes full use of J2EE but avoids proprietary extensions, allowing companies to deploy on application servers from IBM and Oracle—or go open source with JBoss, Apache, and Linux. This openness protects against vendor lock-in.

IFS APPLICATIONS 7 ARCHITECTURE AND TECHNOLOGY



Clients communicate through Http(s) to the web and application servers, which in turn access the database instance through JDBC.

The database runs both the storage tier and the application core business logic objects. The J2EE application server runs the services layer business logic objects, the web/portal presentation tier, and integration services. Both the database and the application server can be run in clustered configurations for extreme reliability and scalability. IFS Web Client and portals can be deployed in the main application server or in a separate web application server. A front-end web server such as Microsoft IIS or Apache can also be used as a complement to the application server.

Clients access the business logic predominantly by using Http (or Https using SSL) protocol. This allows easy passage through firewalls, proxies, and other network infrastructure. Port numbers can be configured. Integrations and customer interfaces access the business logic through the same access providers as used by IFS Applications clients, and thus use the same protocols.

### Windows, UNIX and Linux platforms

IFS' ongoing goal is to offer customers a choice of hardware, operating systems, application servers, or integration technology platforms on which to run IFS Applications. Instead of trying to lock customers into a particular technology stack, IFS works actively to enable customers to choose and change over time. Whereas some companies predominantly use Windows platforms, others rely on UNIX or Linux. Consequently, IFS actively works to integrate IFS Applications

into both Windows and UNIX/Linux environments. This includes the ability not only to run server-side components on either platform but also to provide plug-and-play support for important infrastructure on the respective platform. For example, IFS Applications can use either Microsoft Active Directory, or any LDAP directory for authentication. The front-end web server can be either Apache or Microsoft IIS. Add-on and integration development can be done using Java/J2EE or .NET.

Similarly, IFS sees no need to choose between J2EE and .NET—companies can use both. Packaged applications in combination with continuously changing IT environments mean that most organizations will find themselves operating both J2EE and .NET-based applications. Interoperability between the two is both possible and available. IFS respects these facts, and although the server tier of IFS Applications is implemented according to J2EE architecture, IFS Applications fits well into either environment. This is why IFS Applications supports J2EE technologies such as JSP, JMS, and EJB as well as .NET technologies such as ASP.NET, WinForms®, BizTalk®, and COM+.

## User interfaces

Historically, user interface design has had a technological premise. The interface was designed to work on a particular type of terminal rather supporting the way a user would work. IFS believes that rather than adapting the human to the tool, the tools should be designed to suit human needs.

Creating a good user interface requires design processes that are driven by the task, role, and environment. A good user interface for data entry is different from one for planning or analysis. What works well for engineers in the office might not work at all for technicians in the field. Relying on a single user interface technology for all tasks, roles, and purposes is an outdated principle that does not put the needs of the users first.

Since the 1980s, applications have transitioned from green-screen, through Windows interfaces, to a combination of Windows, web/portal, and PDA clients. This increasing diversity of user interfaces will continue, allowing more people to use applications in more places and on more occasions, and using new devices and terminals. For companies this translates into more value and utility from the same application.

IFS Service-Oriented Component Architecture makes it cost-efficient for IFS to provide a complete set of user interface technologies that suit different users, tasks, roles, and environments. The openness of the architecture also makes it easier to create and integrate other interfaces.

## Optimizing the User Interface

Each company is unique and needs to retain some unique aspects in order to maintain a competitive advantage. The agility of IFS Applications means that companies can optimize the IFS Applications user interface to suit the specific needs of their users and the way they perform tasks. Users can optimize the following aspects of the IFS Applications user interface:

- Tailor screens by moving, hiding, reordering, resizing fields, columns, and tabs.
- Standardize company practice by defining additional fields as mandatory or read only.
- Streamline navigation by customizing the navigator and adding or removing menus for process flow navigation.
- Simplify screens by hiding menus, commands, fields, columns, tabs that are not relevant.
- Boost productivity by creating predefined queries, reports, and page behavior.

All of the above possibilities are available in both web and Windows interfaces. Optimization is done by end-users or super-users in real time directly in the application—without the need for separate and hard-to-use tools. The optimization settings are stored in a user profile. Since each user has both a personal profile and a default profile, super-users or administrators can provide a “base optimization” for a group of users, which they can then further optimize according to their personal preferences. The rights to optimize the user interface can be turned off or granted only to some users.

By more effectively guiding a user to task completion, an optimized user interface not only reduces training needs, but can also reduce stress and increase user satisfaction.

## IFS Web Client and Portal

The IFS Web Client and Portal presents an intuitive and easy-to-use interface to users with a feel that is truly that of the Internet. Portals provide a role-based overview, tailored for each individual. From the portal, users can open detail pages to perform business activities.

The IFS Web Client and Portal interface is a true web interface made up of HTML and JavaScript™ only—applets or other plug-in technologies are not used.

IFS APPLICATIONS 7 ARCHITECTURE AND TECHNOLOGY



Supported browsers include Internet Explorer, Netscape, and Mozilla Firefox. Straightforward management of security and deployment makes it easy to extend access to employees, customers and suppliers.

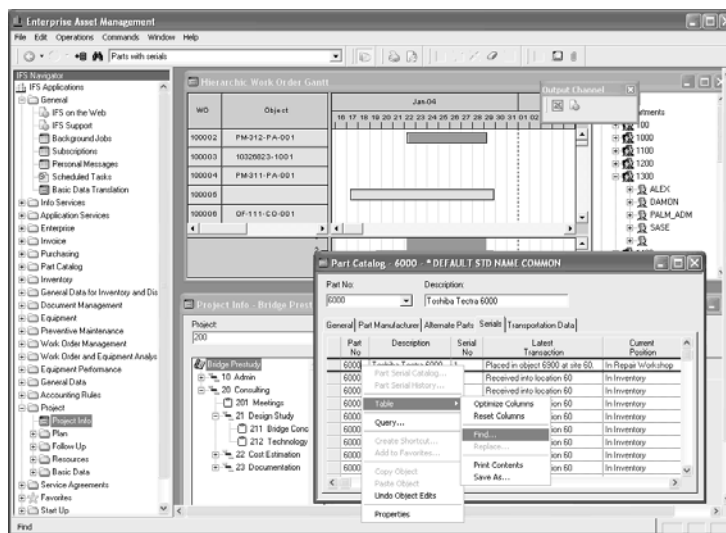
Through the Java Portlet standard (JSR-168) the IFS portal offers standards-based interoperability with other portal servers such as IBM WebSphere Portal

IFS Web Client runs in supported J2EE-compliant application servers and can be deployed behind optional proxies, Apache, or IIS web servers. Authentication of users can either be handled by the web server or performed through the Java Authentication and Authorization Service (JAAS) against any login technology supported by the application server used. All application servers supported by IFS can use Active Directory or LDAP for authentication. In addition IFS provides a login module that allows the Oracle user database to be used for authentication.

## IFS Windows Client

IFS Windows Client offers the richest user experience, with the most extensive optimization and personalization features, rapid data entry, and rich interaction. Integration with Microsoft® Office® provides single-click transfer of data to applications such as Microsoft® Excel and Microsoft® Word. Interactive graphics objects visualize complex data at a glance.

IFS Windows Client can be deployed in a client/server environment, or in a zero-install, web-based environment with Windows Terminal Server (WTS) and/or Citrix software.



## IFS Mobile Client

IFS Mobile Client seamlessly makes information and services available to employees wherever they are located. Offline capability allows IFS Mobile Client to be used with or without a network connection. When offline, synchronization can be performed continuously in the background, or at the user's request. IFS Mobile Client can synchronize and communicate over any IP network supported by the device, including GPRS, WiFi, Bluetooth, Cradle, or regular LAN. Communication is encrypted for security and optimized for performance over the slowest of network links.

Technology lock-in is avoided through the use of Java 2 Micro Edition (J2ME)<sup>™</sup> technology on the device side, and J2EE technology on the server side. This creates device independence and allows IFS Mobile Client to run on popular devices such as the Pocket PCs and ruggedized laptops used today, as well as on new devices emerging tomorrow.

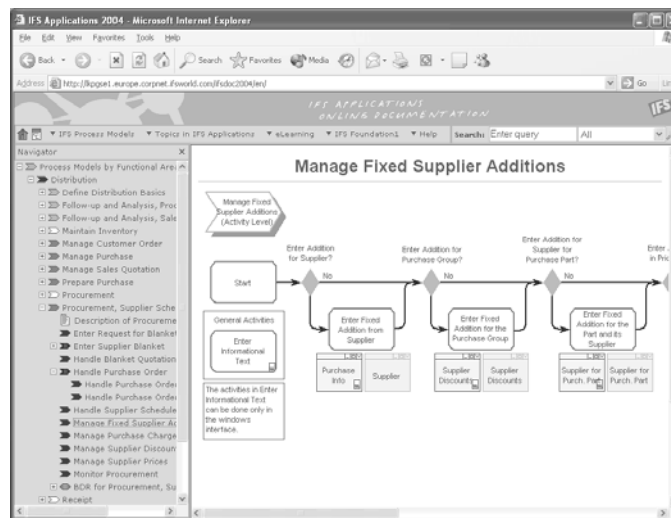


Traditionally, mobile solutions have been complex and costly to set up and manage. Because IFS Mobile Client communicates directly with the services in IFS Applications—without the need for intermediate synchronization software and servers—the entire system integration project normally required for mobile solutions is eliminated. Management is also greatly simplified because there are fewer places where things can go wrong and there is no duplicate administration.

Unique features such as multiple users per device, multilanguage, multicurrency, automated installations and upgrades, and encrypted local storage make IFS Mobile Client fit for use in large deployments.

### Integrated learning

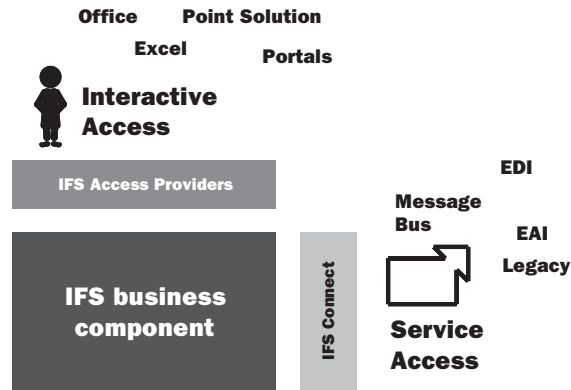
The web-based documentation and training material is integrated directly in the applications. Users jump straight to the relevant parts of the materials, where process models and instructions guide them to the best way of using the applications. Links in the instructions and models take the users to the right places in the application to perform tasks. Detailed explanations of the content and purpose of any field is only a mouse click away.



## Integration, web services

Technology continues to drive a wave of integration. New industry standards enable applications to communicate and work together. Internal systems are linked to each other, to extranets, and to the Internet. Business processes are expanding to include multiple organizations in a networked business community.

For applications, integration is about making independently designed software systems work together. IFS distinguishes between two main categories of integration, driven by different needs. The first category gives users access to IFS Applications functionality using other presentation tier applications than those supplied by IFS. For example,



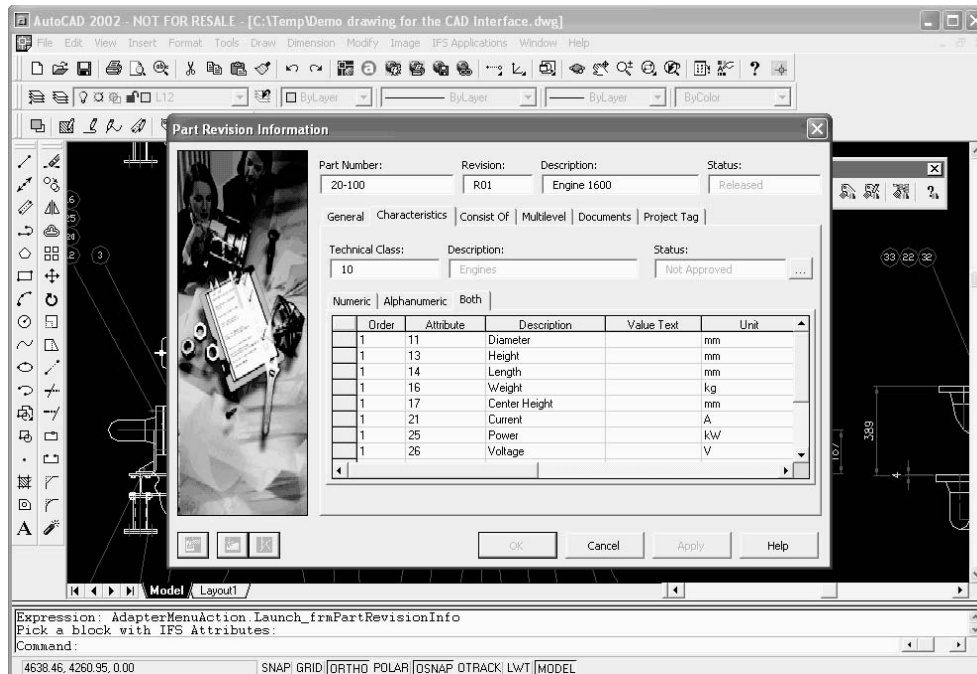
a user might want to build an Excel macro that integrates with the financials solution or an Intranet web page that retrieves certain customer information. The second category is what is usually assumed when talking about integration in the context of business applications. This refers to sending business documents and transactions between systems, to and from customers and suppliers. EDI and EAI fall into this category.

IFS supports the first category of integration with IFS Access Providers—allowing integration of IFS Applications business logic with other clients regardless of the technology with which they are built. The second category is supported through IFS Web Services and IFS Connect.

### Presentation tier integration—IFS Access Providers

For presentation tier integration, the key factor is flexibility. IFS believes in making integration easy regardless of which technology, platform, or tool set is used. This is why IFS has chosen to support both .NET and Java platforms and tools for integration development.

IFS Access Providers are small API libraries that work similarly to ODBC/JDBC drivers. However, rather than talking to the database, they communicate with the application server. They also encapsulate and handle the rich client-server semantics used by IFS Applications. For example, the access providers handle user logon, error messages, and retry/resend logic.



This AutoCAD integration with IFS Applications engineering functionality was created using AutoCAD's built-in VBA scripting language and the IFS COM Access Provider.

IFS Access Providers have been designed to optimize WAN network traffic. The protocol used between the access providers and the server is http, so it can be used across internal and external networks, with full router/firewall security. Http over SSL (https) is also supported. Data sent are compressed by default, drastically reducing bandwidth requirements.

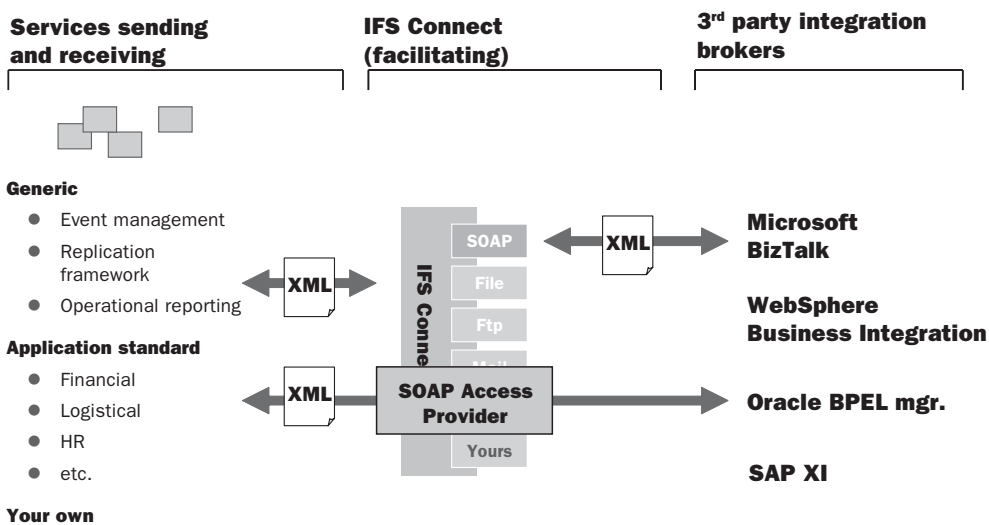
IFS Applications includes access providers for .NET, Java, and SOAP. The SOAP access provider allows any activity or service in IFS Applications to be invoked as a web service.

### Integrating processes and data flows (EDI, web services)

IFS Applications includes a number of services that send or receive information. Some are generic, such as the reporting framework or replication functionality. Other services, such as order confirmations and invoices, belong to specific business components. With IFS Applications, all services sending or receiving information do so using XML natively. This means that IFS Applications is ideally suited to be integrated with other applications that also use XML and web services technology.

Despite the popularity of web services in newer applications, many systems in use still require other ways of integration. These are also supported. Other formats (e.g. another XML format or a delimited file) and different protocols are provided using an integration broker.

IFS Applications comes with its own lightweight integration broker, IFS Connect. Because of the open architecture, IFS Connect can easily be combined with 3rd party integration brokers, EAI and EDI software. Or the 3rd party broker can be used on its own, accessing IFS Applications web services directly through SOAP.



### IFS Connect

IFS Connect is IFS' own lightweight integration broker, designed for XML and web services. It provides additional integration value, including:

- Content-based routing of inbound and outbound messages in one-to-one and one-to-many configurations.
- Master/slave configuration for central routing and administration.
- XML style sheet (XSLT) or Java-based message transformation.
- Support for enveloping (the packaging of messages inside other messages).
- Message store-and-forward and archiving.

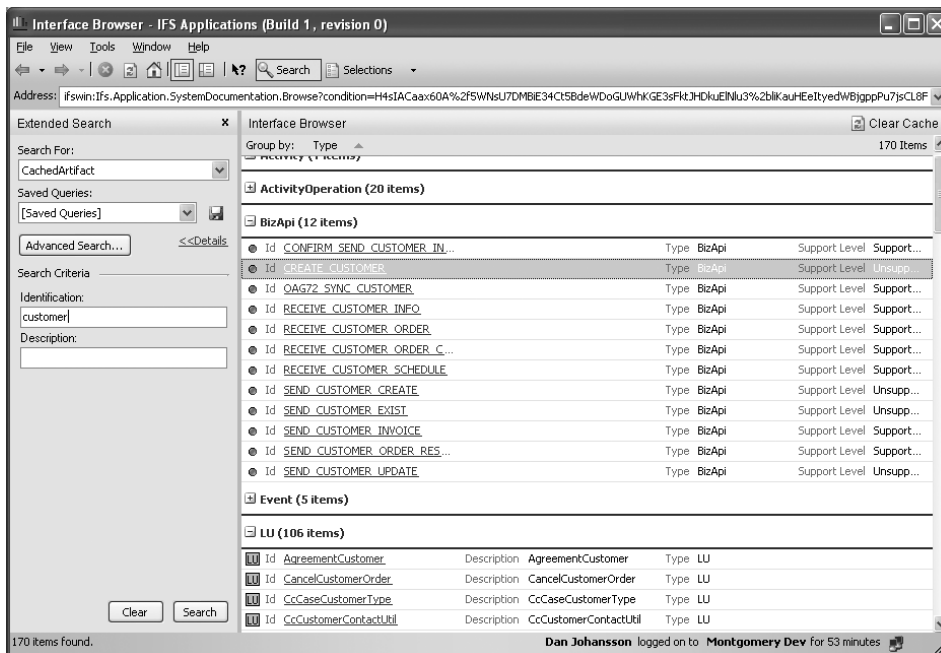
- Ready-to-use connectors based on Internet communications standards, including SOAP, http, https (with SSL), mail (SMTP/POP3), Microsoft BizTalk Server, IBM WebSphere MQ, FTP, and file transfer.

In addition to the integration capabilities, IFS Connect also extends other parts of IFS Applications with additional functionality. For example:

- Business event notifications using mail, SMS, and other IFS Connect connectors.
- Electronic distribution of business document reports as XML or ready-to-view PDF documents.
- IFS Applications replication traffic carried over any protocol supported by IFS Connect.

### Interface browser

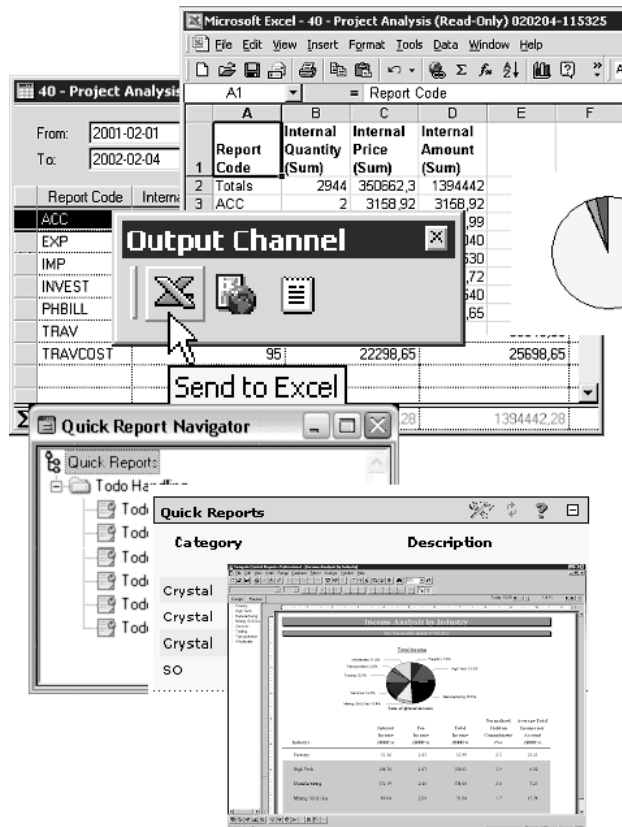
In all integration work, access to technical specifications for services, interfaces and messages is critical to ensure rapid and successful integration. In particular, for web services the use of XML schemas (XSD) to describe message structure and content is a significant productivity enhancer.



The interface browser is an interactive tool allowing an integrator to browse the technical API documentation for IFS Applications. It contains listings and search capability for all IFS Applications core objects, web services, events, information access layer objects, and XML-enabled reports. For each of these, the technical API specification (function names, syntax, data types, etc.) is presented, with XML, XSD, and WSDL documents where applicable.

## Reporting and analysis

IFS Applications has been designed on the principle that access to information viewing, reporting, and analysis functions should be available directly within the application. This allows more users to benefit from up-to-date and accurate information. The principle is apparent in the support IFS Applications provides for everything from the most basic reporting to OLAP analysis, scorecards, and key performance indicators (KPIs).



Considerable “micro-reporting” capabilities are built directly into the IFS Applications user interface. For example, basic group and sum reports can be created directly in the application screens. With the push of a button, the information can be transferred to Microsoft® Excel and other applications for further analysis.

More advanced reports are created using the ‘quick reports’ concept. Report designers can use either plain SQL or Crystal Reports to predefine reports. Users then select, run, and view those reports directly from within the IFS Applications user interface.

### Transforming data into information

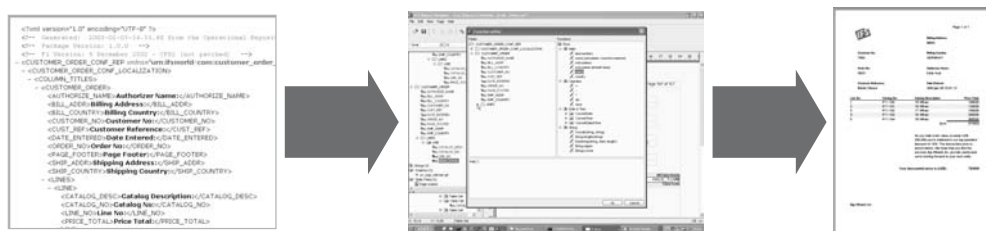
Translating the volumes of transactions—the lifeblood of business—into higher-level business information to make timely, accurate decisions is challenging. IFS Applications answers this challenge with the IFS Information Access Layer (IAL) and IFS Business Performance™.

The information access layer is a set of information components designed to make key information readily accessible. IALs are business-focused and can span multiple applications and data sources to facilitate integrated business reporting and analysis. IALs can be configured to access live transactions for up-to-the-minute reporting or can transfer information to a data warehouse.

IFS Business Performance builds on this concept by supporting management reporting and advanced analysis. IFS Scorecards are used to define key values that can be measured and published throughout an organization using the IFS web portal. OLAP cubes allow users to “slice and dice” through volumes of data to determine trends and support key decision-making.

### Operational reports—IFS Report Designer

IFS Applications has a powerful architecture for operational reporting, i.e. printing of business documents. When a report is ordered, manually or automatically, the data are collected in an XML document. An XSL/FO layout is applied to the data,



Business document reports in IFS Applications are created as XML and printed or previewed as PDF documents

producing a PDF file, which is used for on-screen preview as well as printing and archiving. This architecture blends paper-based and electronic business communication. As well as being printed, any business document report can be sent electronically as an XML message—a convenient function in EAI scenarios.

Companies can easily customize the layout of invoices, order confirmations, and other business documents with IFS Report Designer, a WYSIWYG (what you see is what you get) page designer specially designed to work with XML input data.

## Security

Business applications contain vast amounts of information that is critical to your business, and much that is sensitive or secret. Protecting this information from unauthorized access, tampering, destruction and other malicious behavior remains imperative.

A decade or two ago, IT security was much about controlling system access and backing up data to prevent information loss in the event of system failure or physical destruction such as fire. With the growth of local area networks, wide area networks and later the Internet, focus shifted toward network security. Organizations have since run a tight race against intruders to install firewalls, encryption, virus scanners, and other technologies to protect corporate networks and resources from penetration and sabotage.

As networks are becoming more secure, intruders turn their attention to the applications that run on them. Recent years have seen waves of e-mail viruses and numerous penetrations of well-known web sites. It should be expected that sooner or later similar attention will be paid to business applications. Authorities are also turning up the regulatory pressure on fraud prevention and accountability. Legislation such as the Sarbanes-Oxley Act (SOX) puts a spotlight on the ability of business applications to support segregation of duties, logging, and non-repudiation.

IFS Applications is built on the principle of “secure by design and secure by default” to prevent application vulnerabilities. Security is enforced at the architecture and framework levels, minimizing the risk of vulnerabilities being introduced through the oversight of individual developers. In addition the Foundation1 platform provides a rich set of security services and tools leveraged by IFS Applications and IFS’ customers to implement appropriate security practices.

When it comes to network security, IFS firmly believes in the use of widespread and proven security solutions over home-made “security by obscurity” technology.

IFS also believes that security concepts and the underlying application architecture must be easy to understand and consistently implemented to enable organizations to properly configure the right security. In IFS Applications security is built in—not an afterthought.

### Secure by design and secure by default

To preempt security vulnerabilities, IFS Applications is “secure by design and secure by default”.

Any designs that might affect the security properties of IFS Applications are reviewed by security experts. Questionable designs are rejected in favor of designs whose security implications are easily understood and allow a strong security regime to be implemented. IFS Applications is also designed to prevent exploitation of vulnerabilities that are known to potentially exist in business applications. For example, IFS Applications has built-in protection against SQL injection, session theft, cross-site scripting, and other common vulnerabilities presented by the Open Web Application Security Project (OWASP) and others. Secure by design also means minimizing the damage should a system be penetrated. With this in mind all sensitive data in configuration files are encrypted, so if a web server is compromised, an attacker will not find plain-text passwords or similar that would help further the attack.

Because not all organizations have the same security requirements, there is sometimes a tradeoff between the highest security, ease of use, and cost. Many aspects of IFS Applications can be configured with different implications for security and convenience. For such configurations, the more secure option is always default. This means that initially IFS Applications runs in a “clamped down” mode, and it is up to each organization to enable the additional features or configurations.

### Authorization and segregation of duties

IFS Applications uses a role-based authorization system, which allows clear segregation of duties, including administrative duties, between users. Depending on the duties to be performed, a user is assigned one or more permission sets. Each permission set details exactly what information and functionality may be viewed, updated, or used. Permission sets can also include other permission sets, making it possible to create rights structures of arbitrary depth.

Strong security while maintaining ease of use is achieved through server-enforced permissions with an adaptive user interface. As permissions are set, IFS Applications also grants and revokes the physical software objects, such as database views, that correspond to the permissions. This assures that permissions are

enforced even if users access the database directly using 3<sup>rd</sup> party tools. The user interfaces adapt to reflect what permissions the user has, hiding screens, fields, menu items, etc. that are not available to the user. This helps users focus on the duties they are supposed to perform without getting distracted by things they cannot perform.

A built-in history log function is available to track any modification or removal of data done by users. The log stores information about who made the change, when and where, and records old and new values. Together with security checkpoints that force users to re-authenticate themselves when “electronically signing” certain transactions, the history log is a powerful tool for enforcing accountability and non-repudiation.

### Open and flexible network security

The most important aspect of network security is the use of well-known technologies that have been proven in real-life applications over long periods of time. IFS Applications is built using established technologies with known security properties, including Oracle database, J2EE application servers, Apache and IIS web servers, Active Directory, LDAP, JAAS, Http, SSL, and PKI. Because IFS Applications relies on standards, it can be used with network level security solutions such as firewalls, proxies, and hardware security modules (HSM). De-militarized zone (DMZ), hardened perimeter defense, and other firewall strategies can all be used.

IFS Web Client and all integrations leverage JAAS for user authentication, which means that IFS Applications can leverage login modules provided with the application server used (IBM WebSphere, Oracle Application Server, or JBoss), as well as compatible 3<sup>rd</sup> party modules. In addition IFS also provides an optional login module for the Oracle database, allowing Oracle database user accounts to be used also for authentication of web users.

Grant Permission Set to User		APPROVE_EQUIPMENT_AQUISITION	APPROVE_EQUIPMENT_LOSS	CONDUCT_EQUIPMENT_INVENTORY	MAINTAIN_EQUIPMENT	REPORT_EQUIPMENT_LOSS
Permission Sets						
Users						
+ Department Managers		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>			
- Jacques Villeneuve			<			
Magne Salesman			<			
Engineers						
Scott Epping			<	<		
Scott McGraw			<			
Terje Oftedal			<			
- Alain Leveaux					<	<
Anders Blom					<	<
Gavin DeGrave					<	<
Harald Svensrud					<	<
Ingrid Argenius					<	<
Mechanics						
Jakob Heinemann					<	<
Perry Como					<	<
Per Öquist					<	<

Security between servers, for example from IFS Connect or IFS Mobile Server to the application server, uses the same unified security model as normal users—each server needs to log on as with a user account present in the chosen directory (Active Directory, LDAP, Oracle). This provides effective protection against forged servers.

## Tools

To maximize the return on a business application investment, it often makes sense to optimize the application to the specific needs of a particular organization and implementation. Such optimization may involve a broad range of activities, including optimization of the user interface and reports, configuration of software options, and development of software extensions, integrations or customizations.

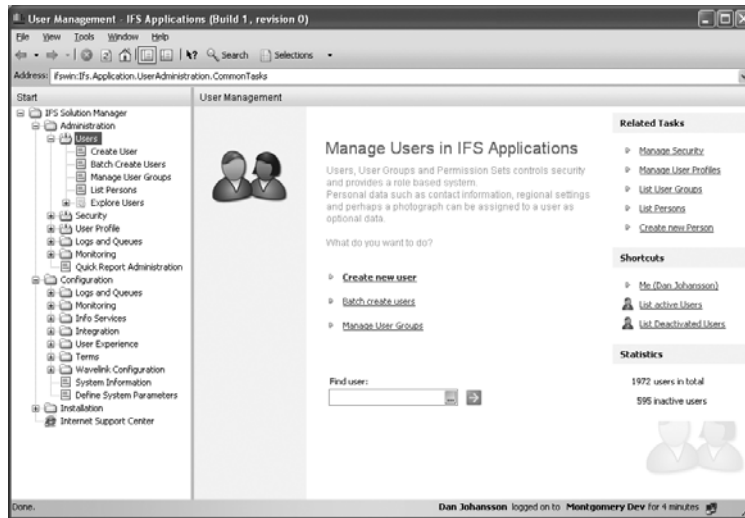
The IFS Foundation<sup>1</sup> platform contains tools that support four optimization aspects of an IFS Applications solution.

- **Software installation.** Initial installation and setup of software, plus subsequent installation of additional options and software updates.
- **Solution configuration.** Setup and configuration of services such as web-service-based integration, event management, history logging, and data replication. May also involve the production of company-specific report layouts, data migration, and similar.
- **Application administration.** Regular administration and management including activities such as user account and profile management, security administration, log and queue management.
- **Custom development.** Development of custom software or integrations that access IFS Applications, or of additional components for the application core, services layer, and presentation layer of IFS Applications.

## IFS Solution Manager

At the heart of the tool set is IFS Solution Manager. Included with all implementations of IFS Applications, it is a powerful tool for administration, configuration and selected installation tasks. IFS Solution Manager has been designed to make the most commonly performed tasks especially easy, while allowing further exploration and analysis.

## IFS APPLICATIONS 7 ARCHITECTURE AND TECHNOLOGY



IFS Solution Manager has been designed to make the most common tasks especially easy to perform

IFS Solution Manager is also an important tool for achieving regulatory compliance. It allows segregation of administrative duties so that, for example, you can make sure that administrators of user accounts cannot delete the logs that track permissions and separation of duties.

In addition to IFS Solution Manager, there are specific tools for further aspects of solution configuration, including:

- IFS Report Designer—an easy-to-use “what you see is what you get” report designer. It allows the creation and modification of layouts for order confirmations, invoices, and other business documents.
- IFS Data Migration—the preferred tool for migrating data from legacy systems into IFS Applications. IFS Data Migration facilitates data loading and cleansing. All data are loaded via IFS Applications business logic to ensure the integrity of the result.
- IFS Software Update Assistant—assists with impact analysis and installation of software updates and patches, as well as re-application of customizations on updated or patched code.

## Custom Development

IFS Applications is constructed using standard development tools and technologies with open interfaces. This creates many opportunities for developers to integrate other applications, write new applications, and extend features within IFS Applications.

The business logic in IFS Applications is contained in a number of software objects grouped into business components. For the purposes of integration, add-on development, reporting, etc., this logic can be programmatically accessed through Application Programming Interfaces (APIs). These APIs are typically accessed from within development environments such as Visual Studio and Eclipse, from macros in products such as Microsoft Word and Excel, from reporting tools, or directly from business logic in other applications. IFS Service-Oriented Component Architecture allows IFS Applications APIs to be accessed in an easy and flexible manner from virtually any development tool or application.

Custom development for IFS Applications is roughly grouped into three areas:

- Programmatically accessing and invoking existing APIs in the services layer or application core layer.
- Development of new, or customization of existing, services in the services layer (middle tier).
- Development of new, or customization of existing, business logic in the application core layer.

Programmatic access to existing services and application core objects is facilitated through IFS Access Providers. This opens up the wealth of IFS Applications for integration into, or development of, entirely new client applications. Access providers facilitate client-to-business-logic connectivity. IFS provides access providers for .Net, Java/J2EE, COM and SOAP, allowing corporate developers to work with any preferred development tool based on Java or .Net, including Microsoft Visual Studio, Eclipse, NetBeans, Oracle JDeveloper, and many others.

IFS supports the development of service layer objects with IFS Developer Studio. Based on the open source NetBeans IDE, IFS Developer Studio adds modeling, code generation, and productivity support for the development of web services, activity services, XML reports and other service layer objects. IFS Developer Studio leverages IFS Applications design models, meta-data, and frameworks to boost

productivity, design reuse, and quality when interacting with IFS Applications core layer. Services developed with IFS Developer Studio become instantly callable from access providers, or through SOAP.

Both the access providers and IFS Developer Studio are used internally by IFS. For example IFS Web Client and Portal uses the Java access provider to access business logic, and all service layer objects included with IFS Applications has been developed with IFS Developer Studio. This ensures that there are no limitations or “hidden APIs” that are available only to IFS, but not to corporate developers.

Development or customization of business logic in the application core layer is the most advanced form of development. This development requires licensing of the entire Foundation1 platform.

## Sizing and scalability

Always working to lower costs of ownership, IFS performs sizing and scalability research to ensure that IFS Applications performs well and fully uses available hardware resources. In addition to routines and guidelines used in all development projects, IFS has a separate benchmarking team focusing solely on tasks related to performance and scalability. Apart from a major involvement in development projects, the team is responsible for maintaining an up-to-date view of IFS Applications on different platform configurations. This is done in close cooperation with major hardware manufacturers, both through continuous testing in in-house laboratories and by visiting test facilities at the vendors for large-scale testing. Tests are regularly performed to assess the impact of:

- New releases of IFS Applications
- Introduction of new hardware
- New releases of 3<sup>rd</sup> party operating systems, databases, application servers, and other 3<sup>rd</sup> party software

The results of the tests are analyzed and fed back to the development teams.

## IFS Applications sizing guides

A correctly sized system is vital to achieving low cost of operation. An under-dimensioned system will slow business down, whereas an oversized system will waste expensive hardware resources. To help customers correctly size their IFS Applications installations, IFS supplies an interactive sizing guide, available online to all IFS subsidiaries and partners globally. The guide covers server and network resources as well as sizing of optional terminal servers. It incorporates all the knowledge gained from continuous benchmarking and test activities with business partners and hardware vendors.

A COMPREHENSIVE SIZING GUIDE TO  
IFS APPLICATIONS 7

Database server | **Web server** | Connect server | Print server | Summary

Select platform

- Intel Xeon (Windows, Linux)
- IBM pSeries (AIX)
- Sun Fire (Solaris)
- HP Integrity (HPLUX, Linux)
- HP 9000 (HPLUX)

Hardware manufacturer: HP, IBM, Dell etc.  
Server models: -  
Operating system: Windows, Linux

Show a printer-friendly version of this section...

**CPU requirements**

You will need a server capable of approximately **666 ILU** (approximately 8000 tpmC). To be able to handle that amount of load, we suggest the following server:

Number of CPUs: 2 x 2800 Mhz  
CPU line: Xeon  
ILU: 3208  
Oracle license: Standard Edition

= Required ILU  
 = ILU of this configuration

For a number of reasons, the recommendation above may not be optimal in your particular case. For a list of alternative Intel Xeon servers, [click here...](#)

For details and updated information, visit the following major vendors:  
♦ <http://www.ibm.com> ♦ <http://www.hp.com> ♦ <http://www.dell.com>

**Memory requirements (32-bit Oracle)**

Based on the information you entered, you will need **256 Mb** of memory available to IFS Applications on the server. The memory of the production system will approximately be distributed according to the following model:

- Fixed OS and Oracle memory (200 Mb)
- Server memory occupied by Win clients (0 Mb)
- Server memory occupied by Web clients (0 Mb)
- Installed components (0 Mb)
- Batch and background jobs (56 Mb)

**Disk and storage requirements**

Based on the expected data volumes and timeframes specified, you will need **54 Gb** to store the accumulated data and indexes. The disk layouts suggested below takes both fault tolerance and performance into consideration.

- S.A.M.E. - high redundancy and high performance
- Traditional RAID 0+1 (high redundancy and high performance)
- Traditional RAID 5 (high redundancy and medium performance)

Based on the customer's unique configuration of IFS Applications, the IFS Applications sizing guide provides hardware and network sizing suggestions for multiple platforms.

## About IFS and IFS Applications

IFS develops and supplies component-based business applications for medium and large enterprises and organizations. IFS Applications, which can be implemented step by step, is based on web and portal technology. The solution offers 60+ enterprise application components used in manufacturing, supply chain management, customer relationship management, service provision, financials, product development, maintenance and human resource administration. IFS offers customers an easier, more open alternative.

IFS is a leading global business applications supplier with sales in 45 countries and more than 350,000 users worldwide. The company is listed on the Stockholm Stock Exchange (XSSE: IFS).

If you need further information, e-mail to [info@ifsworld.com](mailto:info@ifsworld.com), contact your local IFS office or visit our web site:

**[www.IFSWORLD.com](http://www.IFSWORLD.com)**

This support offer has been made in order to respond to the requirements of IFS' customers. Since the customers' requirements may be different in some markets, variations of this offer may exist.

IFS and all IFS product names are trademarks of IFS. The names of actual companies and products mentioned herein may be the trademarks of their respective owners. The example companies, organizations, products, domain names, email addresses, logos, people and events depicted herein are fictitious.

No association with any real company, organization, product, domain name, email address, logo, person or event is intended or should be inferred.

This document may contain statements of possible future functionality for IFS' software products and technology.

Such statements of future functionality are for information purposes only and should not be interpreted as any commitment or representation.

©2006 IFS