



TINREAD

DIGITAL LIBRARY SOFTWARE



System Overview



România

TINREAD

SYSTEM OVERVIEW

April 2007

79-91 Traian Popovici, 031422
Bucharest, Romania
Tel: +40 21 321.19.91
Fax: +40 21 323.61.04
sales@ime.ro
www.tinread.ro

CONTENTS

1. Technology	4
1.1. Eclipse.....	4
1.2. Hibernate.....	4
1.3. Web Server	4
1.4. Databases.....	5
1.5. TINREAD.ILS	5
1.6. Client software	6
1.7. Procedures.....	6
2. Operating systems.....	7
2.1. TINREAD.Mini – Standalone	7
2.2. TINREAD.Standard – Multiuser	7
2.3. TINREAD.Expert – Cluster	7
3. Standards	8
3.1. UNICODE and sorting order	8
3.2. MARC formats and FRBR.....	9
3.3. Output formats	9
3.4. Offline data exchange	10
3.5. Online data exchange	11
3.6. Security	11
4. TINREAD.Engine.....	11
4.1. Data containers.....	12
4.2. Data flow	12
4.3. Forms	13
4.4. Data input and validation	13
4.5. Users and user classes	13
4.6. Menus and submenus.....	13
4.7. Lists and Filters.....	13
4.8. Access rights.....	14
4.9. History	14
4.10. Searches and Runtime indexes.....	14
4.11. Reports.....	14
4.12. Translation	15
4.13. System Parameters	15
5. Configuration samples.....	15
5.1. TINREAD.Mini – Standalone	15
5.2. TINREAD.Standard – Multiuser	16
5.3. TINREAD.Expert – Cluster	19

TINREAD

System Overview

1. Technology

TINREAD.Engine is designed to be a flexible, customizable, high performance, fast and easy to use, scalable, robust, secure and modular platform for the fast development of applications for modern libraries. Each of these goals requires the appropriate technology and tools. Java (**J2EE**) and **SQL** technology meet such criteria and were the ones selected for such a complex project. Java Platform, Enterprise Edition (Java EE) is the industry standard for developing portable, robust, scalable and secure server-side Java applications. Built on the solid foundation of Java SE, Java EE provides web services, component model, management, and communications APIs that make it the industry standard for implementing enterprise class service-oriented architecture (**SOA**) and web 2.0 applications. The database connectivity is performed via **JDBC** drivers thus making the software 100% pure Java allowing it to be platform independent.

The most popular application developed with TINREAD.Engine is **TINREAD.ILS**. Because this application is the subject referred in most cases, anytime when no other qualifier is specified but **TINREAD**, we mean the ILS application TINREAD.ILS. The functionality of an ILS or DLS is available in TINREAD.ILS (or simply TINREAD). The generator used for the deployment of this popular application is TINREAD.Engine.

1.1. Eclipse

Eclipse [www.eclipse.org] platform is the development environment used for TINREAD.Engine itself. *Eclipse* is an open source community whose projects are focused on providing an extensible development platform, a vendor-neutral open development platform and application frameworks for building software. *Eclipse* based tools give developers freedom of choice in a multi-language, multi-platform, multi-vendor environment. *Eclipse* provides a plug-in based framework that makes it easier to create, integrate and utilize software tools, saving time and money. By collaborating and exploiting core integration technology, tool producers can leverage platform reuse and concentrate on core competencies to create new development technology. The Eclipse Platform is written in the Java language and comes with extensive plug-in construction toolkits and examples. It has already been deployed on a range of development workstations including Linux, HP-UX, AIX, Solaris, QNX, Mac OS X and Windows based systems. The founding Strategic Developers and Strategic Consumers are Ericsson, HP, IBM, Intel, MontaVista Software, QNX, SAP and Serena Software.

1.2. Hibernate

Hibernate [www.hibernate.org] is a powerful, high performance object/relational persistence and query service used for data modeling in TINREAD.Engine. Hibernate lets one develop persistent classes following object-oriented idiom - including association, inheritance, polymorphism, composition, and collections. Hibernate allows developers to express queries in its own portable SQL extension (HQL), as well as in native SQL, or with an object-oriented Criteria and Example API. Unlike many other persistence solutions, Hibernate does not hide the power of SQL from you and guarantees that your investment in relational technology and knowledge is as valid as always.

1.3. Web Server

Apache Tomcat [<http://tomcat.apache.org>] is the server-side component used in most of TINREAD implementations. *Apache Tomcat* is the servlet container that is used in the official Reference Implementation for the [Java Servlet](#) and [JavaServer Pages](#) technologies. The Java Servlet and

JavaServer Pages specifications are developed by **Sun** under the [Java Community Process](#). *Apache Tomcat* is developed in an open and participatory environment. *Apache Tomcat* is intended to be a collaboration of the best-of-breed developers from around the world. The *Apache* projects are characterized by a collaborative, consensus based development process, and a desire to create high quality software that leads the way in its field.

TINREAD.Engine was designed from the very beginning of the project in order to be a platform independent framework. *Eclipse* and *Hibernate* obey this basic principle and the output of this development tools are Java files containing the application itself. These files can be deployed to users using any other alternative Web Server compatible with J2EE technology. **BEA Weblogic**, **Oracle AS**, **IBM Websphere** are 100% compatible with TINREAD.Engine applications and are alternative solutions for the Web Server component.

1.4. Databases

TINREAD.Engine is a relational database platform and customers have two choices to use the database engine: as an open source or as a commercial database system. The open source choice is PostGresql [\[www.postgresql.org\]](http://www.postgresql.org) and the commercial option is Oracle [\[www.oracle.com\]](http://www.oracle.com).

PostgreSQL is a powerful, open source relational database system. It has more than 15 years of active development and a proven architecture that has earned it a strong reputation for reliability, data integrity, and correctness. It runs on all major operating systems, including Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), and Windows. It is fully ACID compliant, has full support for foreign keys, joins, views, triggers, and stored procedures (in multiple languages). It includes most SQL92 and SQL99 data types. It also supports storage of binary large objects, including pictures, sounds, or video. It has native programming interfaces for C/C++, Java, Perl, Python, Ruby, Tcl, ODBC, and exceptional documentation.

Oracle Database is the first database designed for grid computing, the most flexible and cost-effective way to manage enterprise information. It cuts costs of management while providing the highest possible quality of service. In addition to providing numerous quality and performance enhancements, Oracle Database 10g significantly reduces the costs of managing the IT environment, with a simplified installation, greatly reduced configuration and management requirements, and automatic performance diagnosis and SQL tuning. These and other automated management capabilities help improving DBA and developer productivity and efficiency. Moreover, with the second version (Release 2), Oracle is focused on improving efficiencies and reducing the cost of information management.

1.5. TINREAD.ILS

TINREAD application (TINREAD.ILS) is a flexible solution for libraries. As detailed in *TINREAD – Product Overview*, TINREAD is deployed in three versions, each one covering specific needs for specific markets.

TINREAD.Mini is the standalone application designed for small libraries, like school libraries. This version can be used by one single user and has several limitations regarding the total number of records allowed by the system to be stored. Most of TINREAD.Mini systems use **PostGresql** database for obvious reasons.

TINREAD.Standard is a powerful version of TINREAD.ILS and its targets are medium and large libraries. This is the Client/Server version of the system designed for Multiuser applications. TINLREAD.Standard has the entire standard functionality of a traditional ILS. Many TINREAD.Standard systems in use are using **PostGresql** for database management but also **Oracle** Database can justify its use in such cases. This version of TINREAD can use any of the two databases with no particular differences for system technical performances.

TINREAD.Expert is a modern DLS and you can get with it the very best of TINREAD. The Expert of the family is carefully designed for huge national/international projects involving Information Retrieval. This special version includes the entire functionality of a traditional ILS and adds specific functionality to DLS (Digital Library System). A lot of Expert options are not available in Standard version (ex. FRBR

cataloguing). Content Management, Real Application Cluster and Disaster Recovery components available from **Oracle** makes it the most suitable RDBMS for such projects.

1.6. Client software

Eclipse environment used by TINREAD.Engine helps developers maintain Java files representing the source code embedding the functionality of the applications. The interface of TINREAD.Engine generated applications consists of objects that users interact with at runtime. This client interface can be chosen from various options.

The standard option for TINREAD.ILS is the Web Browser interface. In order to get such an interface Eclipse is added with **Tapestry** [<http://jakarta.apache.org/tapestry/>] framework. Tapestry is a framework for creating dynamic, robust, highly scalable web applications in Java. Tapestry complements and builds upon the standard Java Servlet API, so it is able to work in any servlet container or application server.

On the client side TINREAD users are able to use any graphical Web Browser supporting Cookies & Java Script such as but not restricted to **Internet Explorer** (v. 5.5 or higher), **Netscape** (v. 8.0 or higher), **Mozilla Firefox** or **Opera**.

Tapestry divides a web application into a set of pages, each constructed from components. This provides a consistent structure, allowing the *Tapestry* framework to assume responsibility for key concerns such as URL construction and dispatch, persistent state storage on the client or on the server, user input validation, localization/internationalization, and exception reporting. Developing *Tapestry* applications involves creating HTML templates using plain HTML, and combining the templates with small amounts of Java code using (optional) XML descriptor files. *Tapestry* brings true object oriented development to Java web applications. *Tapestry* is architected to scale from tiny applications all the way up to massive applications consisting of hundreds of individual pages, developed by large, diverse teams. *Tapestry* easily integrates with any kind of backend, including J2EE, HiveMind and Spring.

SWT (The Standard Widget Toolkit) [<http://www.eclipse.org/swt/>] can be alternatively used to add client interface for TINREAD application. SWT is a widget toolkit for Java designed to provide efficient, portable access to the user-interface facilities of the operating systems on which it is implemented. Using this toolkit with Eclipse, TINREAD client software can be easily deployed like native applications for the host operating system, such as **EXE** files (Windows), **ELF** binaries (Linux) or **Mac** OS X applications.

IBM has donated translations from their products (including SWT) to *Eclipse*. This language pack for SWT contains translations for German, Spanish, French, Italian, Japanese, Korean, Portuguese (Brazil), Traditional Chinese and Simplified Chinese. These translations are based on the Eclipse 3.0.1 build but should work with all 2.1 and 3.0 releases of SWT.

The recommended client interface for TINREAD systems is the Web Browser interface. Client “software” can be easily updated and maintained up to date, and different other aspects involving a better management of the system and commercial aspects are also arguments for such an approach. Even so, any other type of client interface/software as mentioned above can be produced.

1.7. Procedures

The process of managing such a complex project as TINREAD.Engine made mandatory the use of quality control procedures and comprehensive technical procedures. The model of TINREAD.Engine project was established using **ISO 9001:2001**, **ISO 15288** and **CMM Level 3** (SEI Capability Maturity Model) procedures [<http://www.sei.cmu.edu/>]. The CMM covers practices for planning, engineering, and managing software development and maintenance. These key practices improve the ability of organizations to meet goals for cost, schedule, functionality, and product quality. The *CMM* establishes a yardstick against which it is possible to control, in a repeatable way, the maturity of an organization's software process. Software modeling was achieved using all levels of **UML** (Unified Modelling Language) [<http://www.uml.org/>] standard starting with the most general description of Modules interaction down to Function lists and their complete description (input, output, functionality).

2. Operating systems

Each version of TINREAD application has its own demands for technical resources. Hardware resources needed are described in *Configuration Samples* Chapter. Software requirements are presented in the followings.

2.1. TINREAD.Mini – Standalone

This Single-user version of TINREAD uses a single computer in order to perform both server and client operations. On the client side, software resources needed are minimal as specified in *Client Software* sub-chapter: just a Web Browser supporting *Cookies & Java*. All well-known operating systems meet such criteria so there is no restriction in this.

On the server side, obviously some supplemental conditions need to be met. The Web Server (usually *Tomcat*), the JRE environment and PostGresql database should run on the target operating system. Most of the operating systems meet such conditions.

Some of the operating systems suitable for TINREAD.Mini applications are mentioned below.

- Linux (Red Hat 9, Fedora Core 3, Debian etc)
- Microsoft® Windows XP®, Windows 2000/2003 Server
- Sun Solaris 10 etc.

2.2. TINREAD.Standard – Multiuser

The Multi-user version of TINREAD requires at least two physical machines (computers): the server and the client. The conditions for the client operating system are the same as the ones explained for TINREAD.Mini for obvious reasons.

Server operating system must be able to run the Web Server (usually *Tomcat*), to support the JRE environment and the database chosen for the implementation (PostGresql or Oracle).

Client OS

- Linux (Red Hat, Fedora, Debian etc)
- Microsoft® Windows 98, ME, Workstation, XP, Windows 2000/2003 Server
- Sun Solaris
- Mac OS X etc

Server OS

- Linux (Red Hat 9, Fedora Core 3, Red Hat Enterprise, Debian etc)
- UNIX (AIX, HPUX)
- Microsoft® Windows XP®, Windows 2000/2003 Server
- Sun Solaris 10 etc.

2.3. TINREAD.Expert – Cluster

Highest performances obtained by TINREAD.Expert need special requirements. This version of TINREAD is the one recommended for large scale projects with heavy usage and critical downtime conditions. Such missions are successfully managed by computer farms joined together using Cluster technology.

Cluster

TINREAD.Engine scalability allows such an intensive use of TINREAD applications. More power is obtained without any structural changes or source code changes. A simple use of Cluster technology supposes to start using the application with one single server. Afterwards, if the load of this initial machine is critical, another server can be added thus making a group of servers working in a cooperative mode to process client requests.

RAC (Real Application Cluster) is the load-balancing solution from Oracle which makes this Database the most suitable for such applications. At operating system level the best certified cluster solution for Linux is Red Hat Enterprise running on IBM Servers making this combination the recommended one.

Client OS

- Linux (Red Hat, Fedora, Debian etc)
- Microsoft® Windows 98, ME, Workstation, XP, Windows 2000/2003 Server
- Sun Solaris
- Mac OS x etc

Server OS

- Red Hat Linux Enterprise
- UNIX (AIX, HPUX)
- Microsoft® Windows 2000/2003 Server

3. Standards

All well-known standards recommended for Integrated Library Systems are fully implemented for TINREAD in order to produce reliable software compliant with best IT&C technologies.

TINREAD is fully UNICODE compliant and adds important features to the simple fact of displaying characters from all languages (i.e. customizable sort order of characters, case/diacritics sensitivity for searches etc). USMARC, UNIMARC or other description standards are used to catalogue documents. All the set of definitions (field, subfield, indicator, help for the usage of the above mentioned) is stored in TINREAD databases and provides the user with helpful information while cataloguing. FRBR descriptions are allowed in TINREAD at two levels: native FRBR or generated FRBR description with a complex set of algorithms transforming standard MARC records. ISO2709 format is used to exchange document descriptions at file level (import/export). Z39.50 and ZING are available as client or server components allowing online exchange of data. Documents are stored in the digital archive in PDF, HTML or XML format ensuring 100% compatibility with third-party software.

3.1. UNICODE and sorting order

The use of this standard solves the problem of diacritical marks and storing data in different languages. There are several fast methods allowing developers to achieve such tasks but only the full and complete implementation of **UNICODE** standard solves the problem for good. All TINREAD.Engine applications are fully UNICODE compliant. Thus, bibliographical records and full-text can be stored and searched using the Cyrillic, Greek, Chinese or Hebraic alphabet instead of transliterating records written in these alphabets.

Storing different character sets involves sorting them properly. General/predefined sort rules are not always suitable for bibliographical needs. TINREAD.Engine has advanced capabilities to customize sort order at run-time. Three approaches are used to achieve such functionality. **Operating system**

capabilities (especially on Linux/UNIX systems) allow System Administrator to modify system wide settings defining the sort order for UNICODE table. More specific options are available at **database** level in both PostGresql and Oracle allowing Database Administrator to chose from a wide set of rules. Modifying these predefined settings is also an option. More advanced options are implemented at **application** level by TINREAD.Engine itself. The mechanism uses two separate fields to store a traditional field. One of them is the display value of the field (the content) and the other is the sort value of the same field. Defining such rules users can modify the sort order as they wish at the very time of inputing data.

3.2. MARC formats and FRBR

TINREAD systems can use bibliographical description standards like **UNIMARC**, **USMARC** or other. Bibliographical database consists of three containers, each one needing to be filled with appropriate definition. These containers are /B, /A and /H. First of all, at system setup these containers are loaded with MARC definitions for the chosen standard. Forms are built based of these definitions, and users can input data. There is no difference in using UNIMARC or **CNMARC** or else, each standard assuming the same operations to be performed at system setup.

State-of-art technology used by TINREAD allows cataloguing and displaying bibliographical data in **FRBR** format [www.ifla.org/VII/s13/frbr/frbr.htm]. There are two levels of accuracy for FRBR descriptions. First level allows cataloguers to input records in MARC format and to display them in MARC or FRBR. Complex and sophisticated algorithms as well as TINREAD database architecture make this conversion possible and accurate. Bibliographical records (i.e. /B container) are divided in three logical areas: **Work**, **Expression**, and **Manifestation**. The /H container is directly connected to **Item** entities of FRBR descriptions. Second level of accuracy is achieved with native, originally FRBR catalogued records – one of the most advanced cataloguing capability of a modern ILS system as TINREAD.

This approach has many benefic implications for TINREAD users. First of all it makes possible the advanced FRBR navigation style through TINREAD bibliographical data. Bibliographical records are well organized and coherent thus avoiding the unpleasant situation of several screen pages with records having the same title (Poetry or Works for example). The ability of automatically converting bibliographical data from MARC to FRBR and vice versa makes TINREAD a powerful tool for training. Cataloguers can use the system to check how a bibliographical description is represented in each format and also to get used to FRBR descriptions, especially for complex ones.

The **entity-relationship** analysis reflected in the FRBR model serves as a useful conceptual framework for a re-examination of the structures used to store, display, and communicate bibliographic data. Further more FRBR offers capabilities of restructuring MARC record formats to reflect more directly the hierarchical and reciprocal relationships outlined in the model. This is a proper model for the so-called "multiple versions" issue. This model could also be expanded in depth to create a fully developed data model that would serve as the basis for the design of an experimental database to assess the efficiency and effectiveness of a database structure patterned on the model.

The data included in FRBR pertain to textual, music, cartographic, audio-visual, graphic and three-dimensional materials; they cover the full range of physical media described in bibliographic records (paper, film, magnetic tape, optical storage media etc.); they cover all formats (books, sheets, discs, cassettes, cartridges etc.); and they reflect all modes of recording information (analogue, acoustic, electric, digital, optical etc.).

3.3. Output formats

Bibliographical data stored by the system is displayed to its users using different formats, each suitable for a certain type of user. Records can be displayed in **FRBR**, **MARC**, **ISBD** or **Labeled** format just by clicking the appropriate button on screen. The Bibliographical References standard **ISO 690:1987** can be also used to format data on screen. This is useful because it allows users to easily make bibliographies directly on screen without the need of running one of the system reports. User searches the records, then selects the interesting ones, then chooses to display them all in bibliographical references format and finally prints the bibliography or saves it to a file or sends it by email.

More complex output formats assume running a report option. Large amount of data can be searched, sorted, indexed, sophisticatedly formatted finally producing a report file in any of the following formats: **RTF**, **PDF**, **HTML**, **XLS**, **CSV** and **XML**. Report generation is fully customizable in TINREAD and interactive tools allow System Administrator to define or customize reports simply by drag&drop. **JasperReports** and the interactive report development tool **iReport** are used for such tasks.

JasperReports is a powerful open source Java reporting tool that has the ability to deliver rich content onto the screen, to the printer or into RTF, PDF, HTML, XLS, CSV and XML files. It is entirely written in Java and is suitable for use in Java enabled applications, including J2EE or Web applications, to generate dynamic content.

iReport is a visual reporting tool based on JasperReports written in 100% pure Java. It can manage charts, images, subreports. Data can be retrieved using JDBC, TableModels, JavaBeans, XML, Hibernate, CSV and it supports output in RTF, PDF, HTML, XLS, CSV and XML.

3.4. Offline data exchange

The Import/Export module of TINREAD.ILS allows bibliographical data to be stored in or to be extracted from system's databases. The two file formats used directly by TINREAD Import/Export operations are **ISO 2709** and **XML**. If TINREAD needs to exchange data with an alien system which is not able to read any of these two file formats, or even if it decrypts the file format it stores the data at a logical level in a different manner than TINREAD, then a file format conversion is needed.

Specialized tools are built for many other file formats such as but not restricted to TXT, DBF, XLS, RTF. When importing data into TINREAD, a conversion of the alien system file format into ISO 2709 or XML is operated first. Finally the resulted file is processed by TINREAD. Exporting bibliographical data to such alien systems not able to decrypt ISO 2709 or XML requires the conversion of the exported TINREAD file to another file format supported by conversion tools as mentioned above.

MARC4J [<http://marc4j.tigris.org/>] package provides TINREAD access to a collection of MARC records in tape format. The MARCHandler interface provides methods to get information about the record label (record position 00-23), control fields (001-009) and data fields (010-999), including indicator values, tag names, subfield codes and data. The character encoding of the original records is preserved. Field data is returned in character arrays. The optional ErrorHandler interface provides methods to handle error messages. MARC4J is also a set of Java programs which allow TINREAD users to convert to and from the **MARC** file format (including full character set conversion) and other formats including **XML**.

The main goal of MARC4J is to provide an easy to use Application Programming Interface (API) for working with [MARC](#) and [MARC XML](#) in Java. MARC stands for MACHine Readable Cataloguing and is a widely used exchange format for bibliographic data. MARC XML provides a loss-less conversion between MARC (MARC21 but also other formats like UNIMARC) and XML.

The MARC4J library includes:

- An easy to use interface that can handle large record sets;
- Readers and writers for both MARC and MARC XML;
- A build-in pipeline model to pre- or postprocess MARC XML using any XSLT processor that supports the JAXP interface;
- A MARC record object model (like DOM for XML) for in-memory editing of MARC records;
- Support for data conversions from MARC-8 ANSEL, ISO5426 or ISO6937 to UCS/Unicode and back;
- Vendor neutral XML support through JAXP and SAX2, a high performance XML interface;
- Support for conversions between MARC and MARC XML;
- Tight integration with the JAXP, DOM and SAX2 interfaces;
- Easy to integrate with other XML interfaces like DOM, XOM, JDOM or DOM4J;
- Command-line utilities for MARC and MARC XML conversions;
- Javadoc documentation.

3.5. Online data exchange

Z39.50-2003 protocol as defined by <http://lcweb.loc.gov/z3950/agency/document.html> is fully available in TINREAD. Also, TINREAD is oriented to **ZING** protocol as defined by The Library of Congress at www.loc.gov/z3950/agency/zing/zing-home.html.

Client capabilities allow multiple, simultaneous Z39.50 calls from OPAC to different outside bibliographical sources. Together with an algorithm used for data de-duplication TINREAD OPAC gains the power of a Collective Catalogue allowing users to simultaneously view many different catalogues (from different libraries or from the same library) grouped together like one single (logical) catalogue.

Server capabilities allow TINREAD systems to be aggregated in complex projects, libraries consortiums, national collective (federal, shared) catalogues or else.

TINREAD is committed to **SIP2** protocol in order for its future releases to easily integrate self-checking equipments and security solutions aiming to fully automate bibliographic units circulation.

3.6. Security

Users authentication is mandatory in order to establish their access rights. This procedure can be performed by password authentication or LDAP integration. Communication between client and server is encrypted using a **x.509** digital certificate according to **ISO/IEC 17799**.

4. TINREAD.Engine

As mentioned in the above chapters TINREAD.Engine is the platform used for the deployment of library specific applications (ILS) or other as shown in the following. This powerful set of tools includes two types of databases: **System Databases (SYSDBA)** and **Application Databases (APPDBA)**. System Databases store all definitions, data and configuration parameters that define a specific application. Application Databases consist of several Containers (sets of databases) defined using a high level of data modeling thus allowing any library specific database to be obtained as a particular case of the generalized form.

Application development means in TINREAD.Engine terms, loading System Databases. This can be, and actually is performed by System Administrator based on the Application Implementation Plan. SYSDBA also stores the behavior of the generated application. Data input methods, validation, flow of inputted data, store and retrieval procedures and also application output are configured by data stored in SYSDBA. The highest degree of customization is guaranteed by these flexible procedures.

The most popular family of applications built with TINREAD.Engine is **TINREAD.ILS** with its three constructive versions: TINREAD.Mini, TINREAD.Standard and TINREAD.Expert. One can consider TINREAD.Mini as a specific version of TINREAD.Engine with particular data stored in SYSDBA. Modifying this data, system is easily changed and the upgraded version of the application will be TINREAD.Standard. Of course, such a procedure assumes that hardware requirements are met for the final product obtained as a result of the upgrade procedure.

TINREAD (TINREAD.ILS) family of applications means properly defining (configuring) data in TINREAD.Engine environment and afterwards just deploying the generated system to its users.

Up to now several standard (ILS) or extended (DLS) modules are defined in TINREAD.ILS: Cataloguing, Circulation, Acquisitions, Serials Control, ILL or else. Other similar or connected modules can be easily defined and used with this integrated system. Development plans include but are not restricted to the following modules:

- Digital References Services (*“Ask the Librarian”*)
- Digital Documents Delivery
- Local Community Services Information
- University Curricula etc

Different configurations of the data stored in SYSDBA allow TINREAD.Engine to perform different functionality as needed. Other integrated systems obtained with the same procedures as TINREAD.ILS, or otherwise said other configurations of SYSDBA databases generate any of the following applications:

- **TINREAD.Archive** – Digital Archive System designed for companies and institutions to build their own digital archives replacing all hard copy documents.
- **TINREAD.Laws** – Legal Database with full-text and analytic capabilities.
- **TINREAD.Products** – Company Products Database used for the management of big companies' products, specifications, attached multimedia materials (brochures), stock quantities etc used for marketing purposes.

A step by step list of operations needed to build an application in TINREAD.Engine is presented in the following sub-chapters. Once again we will make a note that everything is done at run-time without changing the source code.

4.1. Data containers

TINREAD.Engine defines three Containers able to store records descriptions. Two of them allow qualifiers describing the qualities of the records and one allows quantities description. These containers are generalized forms of typical databases used by integrated library systems.

- /B – Bibliographic (quality 1st level)
- /A – Authorities (quality 2nd level)
- /H – Holdings (quantities)

Any of these three Containers can be populated with data thus specifying the bibliographical format used. UNIMARC, USMARC, CNMARC, RUSMARC or any other set of data organized with the same structure can be inputted in SYSDBA containers. This way, fields, subfields, indicators, allowed values and validation rules, associated labels, contextual help or other information are defined at run-time by System Administrator (**SA**). Two or more definition sets can coexist in TINREAD at the same time allowing bibliographical records to be stored in different MARC formats in the same TINREAD implementation.

Data available in these Containers are directly used to build application Forms. Using forms application input is performed and the collected data (bibliographical data) is stored in APPDBA databases.

Further on supplemental Containers are defined by TINREAD.Engine. **Patrons** Container (/P) and **Transactions** Container (/T) are two other fully customizable sets of databases with similar structures as the ones above. System Administrator can manage Patrons registration fields, subfields or indicators needed to control Patrons records and Circulation. Also full control over Circulation transactions is available to SA. The rest of the functionality for Circulation module is achieved using the **Alerts** Mechanism consisting of triggers defined in the system setup procedures.

The other two traditional modules (Acquisitions and Serials Control) needed by an ILS are using two supplemental Containers called /S (**Suppliers**) and /O (**Orders**). Data stored in /S Container defines the full description for bibliographical units suppliers (resellers, distributors, book shops etc). These records defined by /S can be related to records stored in other containers using a similar set of relations (ex. /S connected with /H in order to order items) thus creating records in /O Container. Also an Alert mechanism is available to System Administrator allowing the full control of events in the above mentioned modules. TINREAD.Engine is committed to **ISO 9735 EDIFACT** standard to make the ordering process easier for libraries (data exchanging with suppliers).

4.2. Data flow

Bibliographical records defined by data stored in SYSDBA are connected one to another using a relational mechanism. The relations between fields or subfields are also stored in SYSDBA databases. This is a powerful approach for data modeling at application level and allows also controlling the flow of data from one module to another. Obviously /B (bibliographical records) and /A (authorities) are connected via *many-to-many* relations. Also /B (bibliographical records) and /H (holdings) are connected using just *one-to-many* relations.

The flow of data inside APPDBA databases is fully customizable. Also the flow of data between different departments of the library is performed. Each library has its own specific activities and organization chart and needs a system capable to be perfectly mapped on these. Allowing SA to configure forms, relations, flow of data, user classes, users, menus, submenus, lists, filters and access rights to these objects the complete set of tools is available to model the desired application.

4.3. Forms

SYSDBA Containers store records defining application forms. Based on these definition forms are built dynamically at run-time by TINREAD.Engine. The appropriate code generating labels, check-boxes, drop-downs and other objects is automatically produced by the system.

Each module has forms available for it. Each record type (book, serial, A/V, borrower record, supplier record etc) can be edited with one or many forms. Depending on the access rights assigned to them after the authentication procedure users gain access to one form or another.

There is no restriction in the number of forms attached to a record type populating one of the Containers and also there are not restrictions for the total number of forms available to one user to edit a certain record type.

4.4. Data input and validation

Input data validation is a very important procedure for a robust application. TINREAD validation is performed at three levels. **First**, just by defining MARC formats to be used with the application indicators specify possible values for a field or subfield. According to these definitions the appropriate code defining the input Form is generated at run-time allowing users just to choose from a restricted set of options and not allowing them to input free text. A **second** level of input validation is performed using regular expressions for field validation. This means the input should obey a certain rule described with usual regular expressions syntax. If the free text input doesn't conform to the validation rule user is requested to make the appropriate adjustments. The **third** level of validation for data input is achieved via authority lists control meaning that data is checked against a table filtered or not.

4.5. Users and user classes

TINREAD.Engine allows SA to define User Classes and their access rights to different objects. Each user allowed to use the system is assigned to one or more User Classes. Access rights are inherited from the attached User Classes or can be directly assigned to a particular user. This scheme allows SA to configure application modules and each operation performed by specialized staff members.

Controlling access rights to different area of bibliographical records means controlling bibliographical authority level. Such capabilities are mandatory for the process of shared cataloguing (national wide).

Please check also the *Access Rights* section to see which objects access rights restrictions apply to.

4.6. Menus and submenus

Application menus and submenus are also stored in SYSDBA area. TINREAD.Engine provides an easy to use interface allowing SA to define the whole structure of application menus. The end node of a menu or submenu is associated to a list.

System Administrator can fully customize all of these objects in order to customize the application or simply just operating the translation in an alternative language.

4.7. Lists and Filters

A list allows users to browse or edit data in the associated database. The list has one or more filters that can be selected with a drop-down menu. Filters apply to view only certain sets of records meeting the criteria defined by the filter. For example SA can configure different filters for Transactions database

(Circulation) allowing users to check which borrowers overdue their borrowing period first time (appropriate filter defined) or second time (another filter). Another simple application is to associate filters to Authorities database. Different filters are defined at startup allowing users to check the list of Authors or Editors or Corporate Authors by simply selecting the appropriate filter applied to Authorities Database.

4.8. Access rights

Access rights defined to control access to data stored in APPDBA apply to all objects available to users. One user gains access rights after logging into the system. User rights are inherited from User Class(es) the user belongs to or can be directly assigned to a particular user by SA.

Menus, submenus and lists (end nodes) can be either available or unavailable to a user. One filter or another can also be active or not for a particular user.

When a list is browsed user can have or have not access to:

- just browsing data
- adding records
- modifying existing records
- deleting records

Using a form users gain access to fields. For each field users can be granted with the following:

- no access at all
- view the field
- insert new values for the field
- modify the field
- delete the field (any occurrences but not all of them)
- deleting all occurrences of a field

Please make a note this section does not include access at application level (ex. "maximum loan limit for a certain borrower class" or anything else similar). These values are application specific parameters defined at application setup phase (user classes, branches, borrower classes, loan limits, access limited by IP address, time:hour etc). TINREAD.Engine setup refers especially to a higher level of customization: properties of the **generated application**.

4.9. History

Each transaction involving database modifications is logged. The modified record is stored in a Trash database and can be recovered at any time. Also a timestamp is generated, a description of the operation and the user name are attached to the modified record. Finally the record can be purged from the Trash database if user has the appropriate access rights. These are application level transaction logging capabilities available in TINREAD. Oracle Database has its own transactions logging capabilities at physical level.

4.10. Searches and Runtime indexes

Fields defined in SYSDBA Containers can be used to build indexes thus generating data to search against. This association is done by SA allowing runtime indexes to be built. Fields involved in user searches and their order on the screen are also defined by SA.

4.11. Reports

Complex reports grouping large amount of data can be defined using interactive tools. Data is first of all selected from the database using search forms defined by SA by simply selecting the appropriate search fields and validation rules that apply to them. Then application databases are searched to meet user

specified criteria. The resulted set of records can be edited (adding more records or removing records) and the final set of data is used to perform the report.

Output format is designed by SA using drag&drop procedures with graphical interactive tools. The final result is a file in any of the following formats: **RTF, PDF, HTML, XLS, CSV** and **XML**.

4.12. Translation

Any TINREAD.Engine application can be translated using a very easy to use procedure. System Translator has special access rights and logs into a special account. First he selects the language the translation is based on and then the target language the system is to be translated in.

Using this account he is able to translate any object on the screen just by pressing a keyboard keys combination when the mouse is placed over the object. The original value of the object is obviously displayed on the screen and the translated text can be entered. The new text immediately replaces the old one and user can instantly see the results on the screen.

4.13. System Parameters

Typical system parameters can be configured by SA modifying System Parameters database. Most of the ILS available on the market allows SA to customize the application only by editing this database. The operation is called parameterization and assumes minor or no changes in the structure of the application or its database, data flow or anything else as detailed in the above sub-chapters.

Parameterization is also available for TINREAD.Engine applications but this is the lowest level of customizing and configuring the application.

5. Configuration samples

5.1. TINREAD.Mini – Standalone

Hardware Specifications

- **Processor:** 1Ghz or higher.
- **Total RAM:** Minimum of 256 MB (PostGresql Database, no more than 100,000 records).
- **Hard Disk Space:** 450MB Disk Space on local drive to full install TINREAD.Mini application and all the components needed; additional 1GB will be needed for data storage.
- **Monitor:** SVGA 1024x768 at least.
- **CD-ROM Drive:** 16x or greater.
- **Data Backup:** Backup Unit CD-RW, DVD-RW (with the appropriate capacity).
- **UPS Unit:** An Uninterruptible Power Supply (UPS) to protect your data in the event of a power failure. A surge protector is also recommended if an UPS is not available.

Software Specifications

- **Operating System:** Linux (Red Hat 9, Fedora Core 3, Debian etc), Solaris 10, Microsoft® Windows XP®, Windows 2000/2003 Server. You must have installed the latest available Patches and/or Service Packs from Microsoft. See Microsoft webpage to obtain the latest versions.
- **Database:** PostGresql 8.0 or higher (recommended), Oracle 10g or higher.
- **Web Server:** Tomcat 5.x
- **Web Browser:** Internet Explorer (v. 5.5 or higher), Netscape (v. 8.0 or higher), Mozilla Firefox, Opera.

TINREAD.Mini System Requirements are subject to change.

5.2. TINREAD.Standard – Multiuser

Client computers must be compatible with TCP/IP protocol and must run a graphical Web Browser supporting Cookies & Java Script such as but not restricted to **Internet Explorer, Mozilla, Netscape, Opera**. Hardware specifications for the client side can be checked among Web Browsers specifications but usually this is not an issue since any computer able to browse the web should do.

Server

Hardware specifications for the **Server** must meet different criteria depending directly on system usage. In the followings two typical configurations are presented.

First configuration is designed for a small/medium library with no more than 10 simultaneous users online. *Second configuration* is a typical configuration for medium/large public/university library with no more than 100 simultaneous users.

NOTE: These two configurations are just samples and the system is not restricted to such fixed dimensions. Each system will be considered on a case-by-case basis and the hardware-software needs will be established as a trade costs/performance to better suit customer needs.

Software Specifications - Server

- **Operating System:** Linux (Red Hat 9, Fedora Core 3, Red Hat Enterprise, Debian etc), UNIX (AIX, HPUX), Solaris 10, Microsoft® Windows XP®, Windows 2000/2003 Server. You must have installed the latest available Patches and/or Service Packs from Microsoft. See Microsoft webpage to obtain the latest versions.
- **Database:** PostGresql 8.0 or higher, Oracle 10g or higher.
- **Web Server:** Tomcat 5.x, BEA WeLogic, Oracle AS, IBM WebSphere
- A **Firewall** system on TCP Port 8080 inbound/outbound is needed to prevent unauthorized accesses to the server.

TINREAD.Standard System Requirements are subject to change.

10 SIMULTANEOUS USERS APPLICATION

Application Server & Database Server & Mass Storage

1 x HP tc2120 with 1 x Intel Pentium IV Processor at 2.8GHz

Main Memory = 1GB

Disks = 2 x 36GB (SCSI)

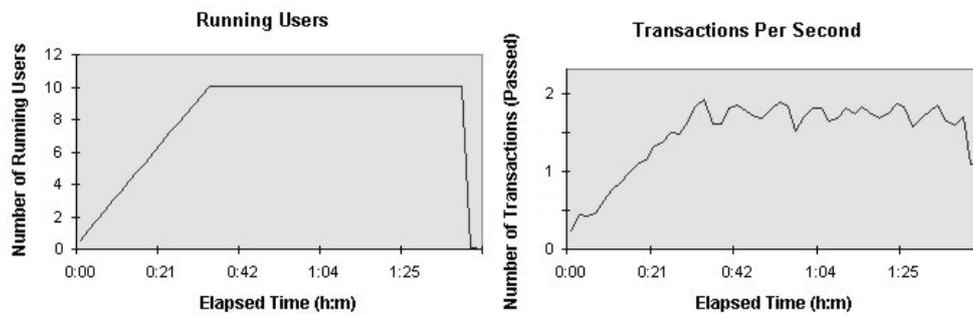
Network Type = 1x Gigabit Ethernet

Technical data

CPUs total no = 1

Max Simultaneous Users = 10

Average Response Time = 0,82s



HP tc2120

100 SIMULTANEOUS USERS APPLICATION

Application Server

1 x Legend SureServerT630 with 4 CPUs x Intel Xeon Processor at 3.0GHz
Main Memory = 8GB
Disks = 1 x 18GB (SEAGATE ST318406LC SCSI)
Network Type = 1x Gigabit Ethernet + 1x 100 BaseT Ethernet

Database Server

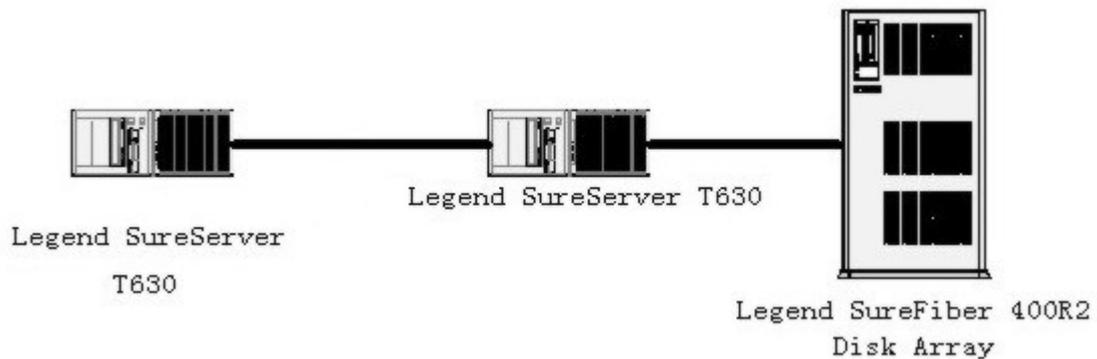
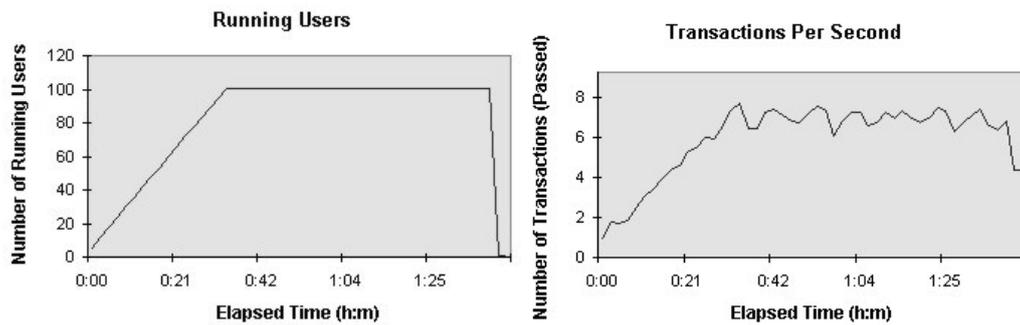
1 x Legend SureServerT630 with 2 x Intel Xeon Processor at 3.0GHz
Main Memory = 3GB
Disks = 1 x 18GB (SEAGATE ST318406LC SCSI)
Network Type = 1x Gigabit Ethernet + 1x 100 BaseT Ethernet

Mass Storage

Needed for large amounts of bibliographical data.
Legend SureFiber 400R2
6 x 18GB (Seagate ST318452FC 15K rpm)

Technical data

CPUs total no = 6
Max Simultaneous Users = 100
Average Response Time = 0,63s



5.3. TINREAD.Expert – Cluster

Client computers must be compatible with TCP/IP protocol and must run a graphical Web Browser supporting Cookies & Java Script such as but not restricted to **Internet Explorer, Mozilla, Netscape, Opera**. Hardware specifications for the client side can be checked among Web Browsers specifications but usually this is not an issue since any computer able to browse the web should do.

Server

Hardware specifications for the **Server** must meet special criteria depending directly on system usage. In the followings two typical configurations are presented.

First configuration is the best choice for a big campus with couple of tenth of branch libraries being able to serve 4,000 active users connected to the system. *Second configuration* applies to very big consortium of libraries with Wide Area Networks and huge amount of data flow. Such hardware configuration provides excellent performance for an environment with 20,000 open connections.

NOTE: These two configurations are just samples and the system is not restricted to such fixed dimensions. Each system will be considered on a case-by-case basis and the hardware-software needs will be established as a trade costs/performance to better suit customer needs.

Mass Storage requirements are calculated depending on client needs and type of materials to be digitized. Extensive details for hardware sizing are available in *TINREAD Hardware Sizing* brochure.

Internet connectivity between the Server and each Client requires a minimum of 32kbps non-latent bandwidth.

Software Specifications - Servers

- **Operating System:** Linux (Red Hat Enterprise), UNIX (AIX, HPUX), Microsoft® Windows 2000/2003 Server. You must have installed the latest available Patches and/or Service Packs from Microsoft. See Microsoft webpage to obtain the latest versions.
- **Database:** PostGresql 8.0 or higher, Oracle 10g or higher (recommended)
- **Web Server:** Tomcat 5.x, BEA WeLogic, Oracle AS, IBM WebSphere
- A **Firewall** system on TCP Port 8080 inbound/outbound is needed to prevent unauthorized accesses to the servers.

TINREAD.Expert System Requirements are subject to change.

4,000 SIMULTANEOUS USERS APPLICATION CLUSTER

Application Cluster

2 x HP9000 rp7400 each with 8 CPUs x PA8700 at 750MHz
Main Memory = 32GB
Disks = 2 x 36GB (internal)
Network Type = 1x Gigabit Ethernet + 1x 100 BaseT Ethernet

Database Cluster

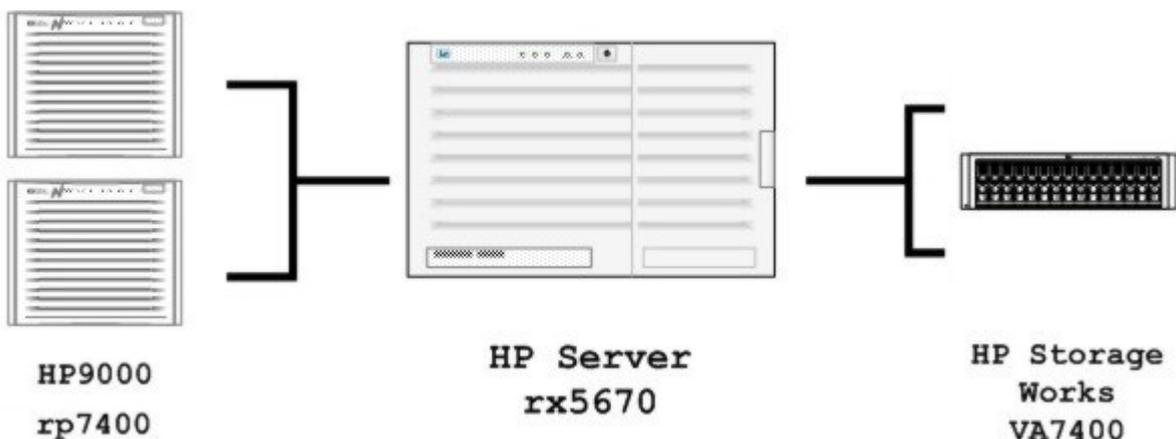
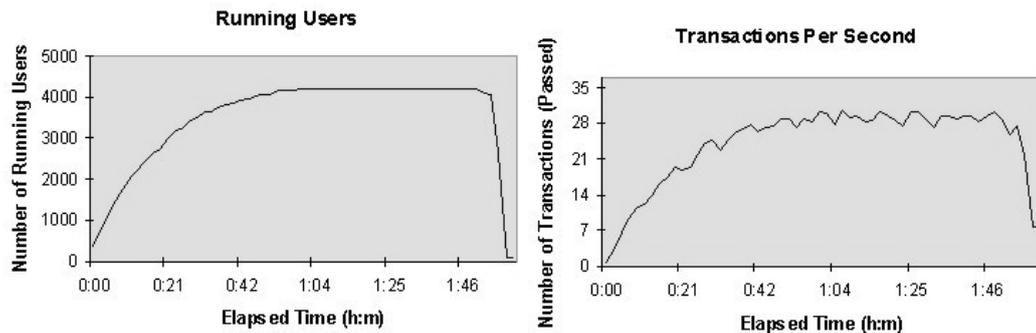
1 x HP Server rx5670 each with 4 CPUs x Intel Itanium 2 at 1GHz
Main Memory = 48GB
Disks = 2 x 36GB (internal)
Network Type = 2x Gigabit Ethernet + 1x 100 BaseT Ethernet

Mass Storage

1 x HP Storage Works VA7400
45 x 36GB

Technical data

CPUs total no = 20
Max Simultaneous Users = 4,000
Average Response Time = 1,51s



20,000 SIMULTANEOUS USERS APPLICATION CLUSTER

Application Cluster

12 x IBM pSeries 650 each with 8 CPUs x IBM Power4+ at 1.45GHz

Main memory = 32GB

Disks = 3x 34GB Ultra3 SCSI (4 servers) 2x 34GB Ultra3 SCSI for other 8 servers

Network Type = 1x Gigabit Ethernet + 1x 100 BaseT Ethernet

Database Cluster

4 x IBM pSeries 655 each with 4 CPUs x IBM Power4+ at 1.7GHz

Main memory = 64GB

Disk controllers = 1x integrated Ultra3 SCSI + 2x Fiber Channel Adapters

Disks = 2x 34GB Ultra3 SCSI

Network Type = 2x Gigabit Ethernet + 1x 100 BaseT Ethernet

Mass Storage

1x IBM TotalStorage FASt700

70 x 36GB disks

Technical data

CPUs total no = 112

Max Simultaneous Users = 20,000

Average Response Time = 0,52s

