



business solutions

The IngleNet Approach to Legacy Modernization:

Technical Overview

The ultimate goal of the modernization project is to create a new IT environment that employs current technology in the areas of database, user interface and development techniques.

*IngleNet, a proven leader in legacy systems modernization, can preserve the value you have invested in information systems and liberate the potential value of your data through the application of our proven and reliable **OpenARMS** modernization process.*

The Inglenet Approach to Legacy Modernization: *Technical Overview*

Today's IT Dilemma

Reduce costs, and improve service! Do more with less! IT departments are under constant pressure to make more effective use of information service resources while reducing costs.

At the same time, system users are demanding more sophisticated software tools together with faster, more powerful hardware. Modern graphical interfaces, distributed processing, increased reliability and scalability are not expected but demanded. While IT departments continue to wrestle with these issues, the list of demands continues to grow: ever-present system maintenance, pressure to implement Internet, Intranet and wireless solutions, Website management, skills upgrading, training, and more.

However, for many organizations, replacing existing legacy systems with custom written solutions or new application packages is simply not an option. The costs associated with replacement strategy may be prohibitive or in many cases is simply not a requirement. In many situations legacy systems work extremely well and serve the information needs of the corporation. Even so, the problem remains that these legacy systems generally run on hardware that is obsolete and/or expensive to maintain. For the most part the software platforms cannot easily take advantage of modern information processing and networking techniques.

Many of our clients have done many things over recent years to try to overcome obstacles and resolve this on their own. Some already have developed procedures to copy the data into SQL tables at night. This makes a non real-time copy of the data that is 24-hours old. They do this just so they can work with the data using modern tools and development environments. What they really want is that their live, transactional data be stored in well-designed (normalized) SQL database.

They don't necessarily want to re-write their applications and many couldn't cope with a code transformation right away.

In cases where the business is not happy with the current application, commercial off the shelf and custom-developed options may be more appropriate. But in cases where the organization relies on the years of business knowledge that went into the application, the need for change is not to replace but to enhance with new capabilities. This is the perfect situation for a modernization project.

Many progressive companies are investigating legacy system modernization as a strategy to provide a way out of their current difficulties. Implementing legacy modernization using open server technology provides companies with an opportunity to leverage their investment in technology and improve information services and reduce costs.

They need the ability to leverage their corporate information in new and better ways for business advantage. This is the first step in a multi-phased approach to legacy system modernization.

With all these internal and external pressures, how can an IT department modernize their systems?

How Does One Modernize a Mainframe System?

More and more Unisys (Sperry) mainframe users are looking to Inglenet for help. Our in-depth knowledge of Unisys mainframes and networking makes us the perfect candidate to modernize your system. Inglenet supports a planned migration of legacy mainframe systems to open systems based on UNIX or Microsoft Windows servers in an enterprise network. Inglenet provides the migration expertise and tools, which permit an organization to adapt mainframe systems to network based server architecture.

The Inglenet compatibility transaction processing software (*TIP Studio™ III*) is written with an understanding of legacy systems and, in many cases, seamlessly enhances the capabilities of the mainframe systems without



business solutions

the need to modify the application code. The reliable and powerful transaction-processing framework IngleNet has provided for more than 25 years to Unisys mainframe customers has been implemented on modern open systems platforms.

IngleNet's **TIP Studio** features may be deployed in a number of ways. The basic configuration consists of a network of PCs with one or more Servers that are used to perform the transaction processing applications. The network must support the standard TCP/IP protocol. In this configuration users will use PCs as workstations. The PCs must be running a supported network version of Microsoft Windows — Windows NT™, Windows 2000™ or Windows XP™.

Application servers may be running on a Windows Server or on a supported UNIX platform. Since all of these components are industry standard, there is a multitude of choice available for vendors and specific add-on items. Other network protocols and associated applications can coexist with **TIP Studio** activities. Multiple servers, distributed servers, wide area network connections, fax servers and other solutions are made feasible by the nature of the network and the open architecture.

An IngleNet solution considers your legacy system, and provides a strategy, methodology and the services for moving to open system server based architecture. With a proven strategy for dealing with modernization issues, IngleNet Business Solutions has successfully managed the migration for many Unisys mainframe customers. In all cases, the projects were carefully planned and executed within the established time frame and budget.

The methodology employed by IngleNet has been refined with the experiences gained from each modernization project — whether the project involved a small System 80 system or a large, enterprise-wide 2200 environment, IngleNet can do the job.

As part of the IngleNet migration methodology, IngleNet has created a comprehensive process and suite of tools called "**OpenARMS™**". This unique process is continuously revised to reflect the latest techniques employed in legacy system modernization.

Our team of professionals has the mainframe experience to understand your legacy systems, and has extensive knowledge of the current suite of UNIX/Linux and Windows based systems to ensure your applications move forward and take full advantage of open system network based computing.

Windows or UNIX/Linux — Use one or both

The choice of server traditionally has been a pivotal decision that sets the direction of all or most corporate computing. However, with today's network technology it is possible to run very different servers at the same time. The design and implementation of the IngleNet products provides a software solution that can be implemented on two very different target platforms: UNIX/Linux and Microsoft Windows. IngleNet believes that the choice of servers does not have to be exclusive; it is reasonable to choose a solution that utilizes the operating system strengths your organization deems critical.

UNIX may be the choice of organizations with very high volume and extensive transaction processing requirements. Modern implementations of UNIX systems provide robust processing capabilities and can be expanded to virtually any capacity.

Microsoft Windows is accepted more and more at the enterprise level. Recent versions, such as Windows 2000 and .NET Servers are proving to deliver on the promise of robust and reliable operations. It is also the choice for organizations that have a need to integrate desktop solutions with products such as Microsoft Office or BackOffice solutions. The modern graphical interface, ease of use and maintenance make Windows an increasingly popular choice. Many organizations are already running mission critical systems such as email or web servers using solutions based on Microsoft Windows Server platforms.

Many organizations may choose to implement a combination of UNIX and Windows servers to provide a number of options to users and IT staff when implementing application systems.

Regardless of your server choice, IngleNet provides transaction-processing software that protects legacy systems and implements open server processing today.

The OpenARMS process in a Nutshell – Data, Interface then Code

The ultimate goal of the modernization project is create a new IT environment that employs current technology in the areas of database, user interface and development techniques. The IngleNet approach to accomplishing this goal is to segregate the entire process into discrete phases where each phase has a specific deliverable with measurable results and benefits.

Attempting to completely overhaul an enterprise class information system in one step is extremely difficult, costly and risky. A single project approach often results in years of effort and expenses being expended before any tangible results are seen. With projects so large, delays and overruns are commonplace and complete failure of a single project approach is not uncommon as the organization's stamina to see the seemingly never-ending project through to completion falters. By breaking the problem down into more easily managed phases, IngleNet has proven its ability to deliver results in a much shorter time frame. The customer is able to enjoy the benefits of each phase as it is completed and utilize these benefits as the project moves on to the next phase.

Reassessment of business priorities can also be taken into account as each phase is completed allowing the organization to determine where new IT investments will have the greatest return. What was thought of as a top priority when the project started may have been usurped by other business requirements by the time the first phase is complete.

Although some flexibility can exist in choosing if and when a specific phase is performed, there is a recommended order to the steps that are executed. In general, the phases of a legacy modernization project are as follows:

- Phase I – The Data Modernization Phase
- Phase II – The User Interface Modernization Phase
- Phase III – The Code Modernization Phase

Why focus on the data first?

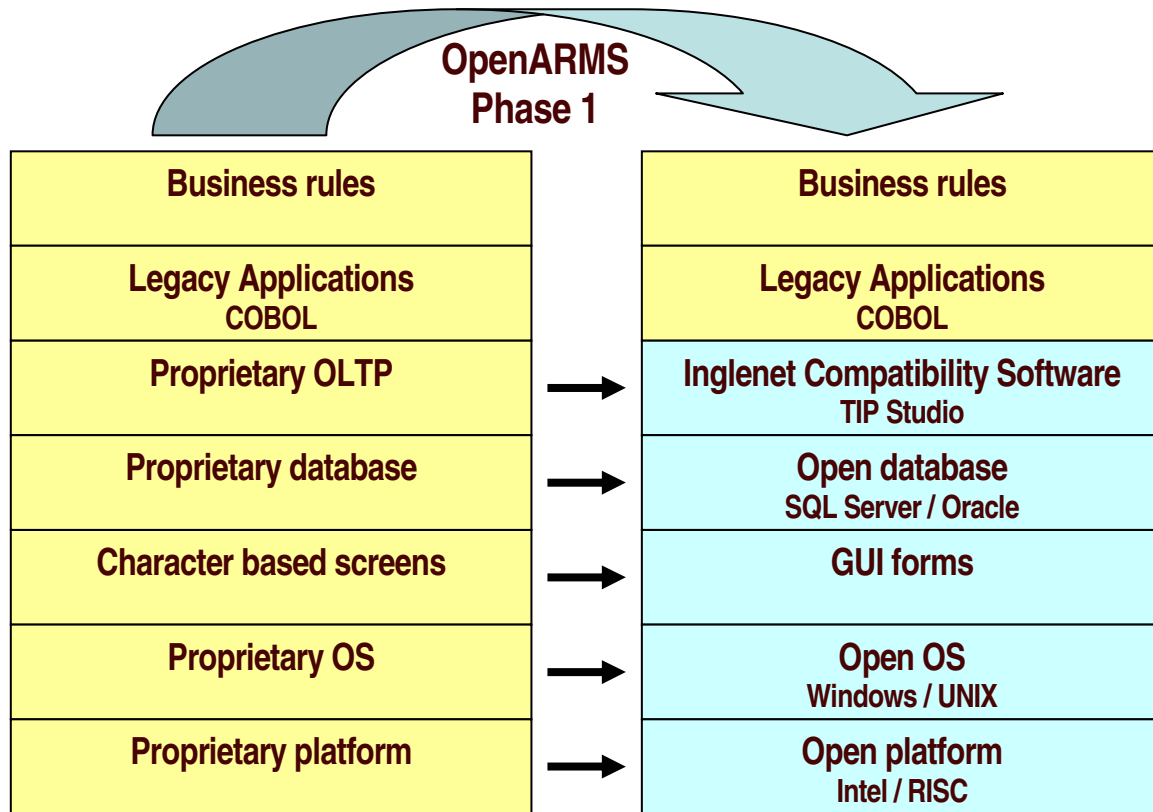
The data is clearly an extremely important and valuable asset of the organization; however, is it truly more important than the code that defines the organization's business rules or the skills that exist in the people who use and maintain the IT systems? It is not the relative importance of the data but its impact on other aspects of the modernization project that determines that it be the focus of the first phase of the project.

If a fundamentally sound data model, that will support the business needs of the organization, is not developed as part of the initial phase then future modernization efforts will be more difficult and less likely to succeed.

To modernize the data, the data must be moved to a newly designed data model hosted by a modern relational database management system on a new platform. This requirement will drive several other activities that all become part of the data modernization phase of the project.

During the first phase, many components of the legacy system are replaced with new components; however the application code remains unaltered by design. By keeping the application code intact, the initial phase of the modernization project is much more manageable and cutover to a new hardware/software platform can be accomplished in a shorter time frame than would otherwise be possible. IngleNet's **TIP Studio** compatibility software components make this possible.

The following diagram illustrates the phase 1 activities of the modernization project.



Throughout this process the operation is protected through extensive testing, parallel operations, and user involvement.

Our initial phase of the modernization project provides the following:

- Develop a new data architecture. A new Architecture simply states the rules for how the data will be managed.
- Analyze and understand the current data storage and retrieval systems, be they DMS, RDMS, flat files, or use some other data storage technique. This is really a consulting assignment that we undertake to understand what the data requirements for the organization are.
- Work with the customer to design a new data storage and retrieval system based on current database technology (RDBMS).
- Provide tools and services (OpenARMS) to transform the data from the old model to the new one.
- Provide interface routines that allow the current legacy applications to run unaltered using the new data model on the new servers.
- Move the existing legacy code to the new platform using the interface routines we have created.

Completing these initial steps already provides tremendous benefits. Data is now stored in a way that it can easily be used by modern reporting and query tools. Development can now build new systems using new tools,

languages and frameworks to further process the data while having their core legacy applications continue to operate the way they always have.

Next phase is to modernize the User Interface.

Modernizing the user interface can provide significant benefits to those who use the information systems. Apart from the enriched experience and all the well-documented ease of use and accuracy features a modern graphical interface provides, the most significant benefit may be one of user perception. When the user interface is modernized to employ current form-based graphical techniques, users have the feeling they are working with a modern information system. The new user interface is also a great way to visibly demonstrate the progress being made in the overall legacy modernization project.

The user-interface modernization phase can either be performed as a distinctly separate phase or it can be combined with the data modernization phase and accomplished during the switch to the new hardware and software platforms. The determination of when this phase is done is usually made based on factors such as; the overall scale of the project, the logistics involved in introducing a new user interface and how much customization may be desired in a new user interface.

Modernizing the user interface is dependant on the kind of interface currently employed by the application programs. If the applications are sending low-level, terminal specific codes such as UTS control codes or DICE sequences, then significant modernization is not really possible. However if the programs are using externally defined formats such as DPS screens, then **TIP Studio™** provides techniques to allow these character-based screen formats to be replaced with graphical forms

Consideration also has to be given to situations where the end user is not a person, but rather another computer. Having one computer communicate with another one through a terminal interface has been a popular technique for sharing information in the legacy environment. This technique, often called "screen scraping" is somewhat fragile as even the most cosmetic of changes in the information displayed by one computer may break this interface. Modern systems have long abandoned these types of techniques in favour of proper programmatic interfaces such as web services and XML data structures. However, these older interfaces have to be preserved as the systems are moved to new platforms. Components of IngleNet's **TIP Studio™** software guarantee that other computer systems connecting into the legacy system running on a new platform will interface exactly as they did when the system was run on the proprietary mainframe. This will ensure that operations can continue uninterrupted. When these interfaces are eventually replaced with a modern one, the networking compatibility software will no longer be required.

The final phase is to modernize the Code

Once the legacy applications have been migrated to a new, open platform the process of modernizing the applications can begin. Generally, there are a few ways this phase of the project can be tackled depending on the priorities and requirements of the organization.

The various approaches to code modernization are as follows:

- **Develop new applications while the legacy applications do the old work.** With IT budgets under constant pressure, it is always a management decision where is the most appropriate place to spend development resources. Organizations may find that the legacy applications are running well in the new environment and are doing what they were built to. In this case, the top priority may not be to replace the legacy applications but rather to add new applications to provide additional solutions that help the organization operate more effectively. These applications may be needed to utilize new technologies that were not available when the legacy systems were built. Applications built to utilize technologies such as the world-wide-web and wireless networks may serve the company better than focusing limited resources on renovating legacy applications that are

performing well. Clearly this is not a long-term strategy, as eventually the legacy code needs to be upgraded to modern standards. However, the upgrading process needs to be factored into the overall priorities of the organization. Key to IngleNet's approach is that by focusing on the data first as a foundation step, you gain the option of building new applications that will operate along side the legacy applications sharing the same modern data model with complete compatibility.

- **Rewrite applications as requirements dictate significant changes.** From time to time in the life of any application suite, changing business requirement will dictate that significant changes are required. The new business requirements may be due to a new direction the business is taking , perhaps legislative changes that must be complied with, or even merger or divestiture activities may have driven the need to change certain applications. Whatever the reason, as these requirements present themselves, they create an opportunity to evaluate if now is the right time to rewrite the application using modern tools, languages and frameworks as a way to incorporate the significant changes required. At this point, due to the work done in data modernization phase, we have already laid down the foundation for rewriting the applications by creating a modern, well structured database that can be used by both legacy and newly created applications. Using this technique, applications are gradually replaced with new ones over time according to the business requirements of the organization.
- **Recode applications manually.** Once the initial phase of the project is complete and the legacy applications are running on the new platform, a process to recode each application in a new language can begin. Recoding applications manually is an expensive activity but if done right, it can produce high quality results that will stand the organization in good stead for the future. Since this activity only replaces current applications with a new one that essentially has the same functionality, organizations may choose to make this a secondary priority that is done when no critical or "must do" tasks exist.
- **Transform applications using an automated process.** If getting the applications converted to a new language is a priority within the organization, then the option of automated transformation may be the process of choice. Tools exist that will convert existing legacy applications originally in early generation languages such as COBOL to modern languages such as Visual Basic or C++. Although the tools cannot transform 100% of the legacy code, they do reach levels of completeness as high as 80 to 85%. Automatically transformed code is likely not going to be of the same quality as hand-crafted code, however the cost of converting to a new language using an automated or semi-automated process is usually significantly less than a manual approach. In this situation, however, we must consider the issue created for post transformation maintenance. The current technical support staff, being well versed in the applications written in the legacy language, are capable of providing support and maintenance relatively quickly. However, once the code has been transformed into a new language, the familiarity that once led to speedy corrections will be lost for a while until the staff is as well versed with the new language, development environment and application code as they once were with the legacy system. For this reason, taking one of the previously mentioned approaches may be more appropriate as a first step. This will give the staff time to become very proficient in the new technology while they replace legacy systems in a more controlled environment. Once the skills in the new development methodologies are at the desired level or the amount of legacy code is reduced to a more manageable level, then an automated code transformation approach may be the most appropriate step to take.

The Offerings

The OpenARMS™ Process

OpenARMS is the process that IngleNet offers to modernize the legacy applications. The **OpenARMS** process consists of a number of services that are provided to ensure that the modernization project will deliver the desired results. These services include the following:

- Planning and Assessment
- Project Management
- Database Design and Implementation
- Data Conversion Services
- Re-Hosting Delivery Services
- Testing Services
- Production Launch Management

In addition to **OpenARMS**, IngleNet also provides **TIP Studio**; software that delivers the runtime compatibility environment needed to allow legacy applications to run unaltered in the new environment. **TIP Studio** is a modular product consisting of many components. These components can be grouped into three major categories as follows:

- TIP Studio Database Interface – data compatibility layer
- TIP Studio OLTP Servers – transaction processing compatibility layer
- TIP Studio Connection Server – network compatibility layer

TIP Studio™ III

What does the TIP Studio Database Interface provide?

The TIP Studio Database Interface consists of tools and runtime systems designed to allow the creation of a new relational database from the existing data stores. One of the first steps in the implementation of TIP Studio Database Interface is the design of the new database.

TIP Studio Database Interface provides tools that parse the existing DMS schema and create a “suggested” SQL schema as a starting point for this process. In addition to the proposed SQL schema, TIP Studio Database Interface also creates additional information that is used to map the new SQL schema to what was available in the legacy DMS schema. This information is generally referred to as the “mapping rules” and is an essential ingredient for TIP Studio Database Interface to provide runtime legacy compatibility. The “suggested” SQL schema can then be modified to better meet the current and future needs of the organization.

Any changes in the SQL schema that affect location or retrieval of information used in legacy applications must also be reflected by appropriate adjustments to the “mapping rules” information. Designing the SQL database to meet the needs of the organization is a very important step in the overall modernization process and may involve extensive requirements analysis and database design skills.

IngleNet can help bring these skills into the organization. We often work together with the customer’s staff to ensure that the database design is fundamentally sound, meets the requirement of the organization and is functionally complete to ensure error-free operation of existing legacy applications.

Once the new database design is complete, the process of moving the database can begin. To do this, TIP Studio Database Interface will generate “unload” programs that are run on the legacy system. These programs

will traverse the existing database and copy the data as unstructured, “flat” images that are copied to the new platform. On the new platform, these “unloaded” files are then copied to the new SQL database by TIP Studio Database Interface generated “load” programs. During this DMS “unload” and SQL “reload” process a number of data transformations take place, including:

- Any data stored using the “field data” character set is translated to ASCII
- Any binary format numbers are converted to displayable numbers
- Any PIC 1 fields are converted to displayable numeric fields
- Any date fields are converted to typed date fields
- Any records with redefined data are converted to a normalized format for a relational database

It is important to understand that the transformation that is done on the data during this process is controlled by the TIP Studio Database Interface mapping rules, so there is complete control over how the data is changed and the format and structure of the new SQL database.

The remaining functionality provided by TIP Studio Database Interface is runtime services that shield the legacy applications from the fact that the data is now stored in a relational database. For both batch and online applications, TIP Studio Database Interface provides interfaces (APIs) that are identical to those that existed on the legacy mainframe. The implementation of these APIs is obviously quite different.

On the mainframe a DML statement, such as a fetch would result in accessing the DMS database on the mainframe. However with TIP Studio Database Interface, the same fetch statement would result in accessing both the relational database and the mapping rules. The mapping rules tell TIP Studio Database Interface where to find the requested data (i.e. what tables and columns are needed) and how to present this data to the legacy application. Status codes are identical to those used on the mainframe so logic changes are not required when the applications are moved to the new platform.

TIP Studio OLTP Servers for Windows and Unix/Linux

TIP Studio OLTP Server is the heart of the transaction processing system. All legacy applications execute under control of the OLTP (On-Line Transaction Processor) server. **TIP Studio** includes implementations of the OLTP server designed for both the Windows operating system and for a variety of UNIX/Linux operating systems. The OLTP Server provides an execution time environment that features:

- high-performance operation
- seamless interface to relational data base management systems (DBMS)
- robust file management including update logging and roll-back, record locking
- distributed transaction and file processing
- application level security
- application programming interface (API) that is easy to use (COBOL CALL level)
- application program isolation to prevent a rogue application program from affecting other user tasks or critical system server tasks.

The OLTP servers also provide application programming interfaces (APIs) that permits legacy programs to operate as if they were still running on the Unisys mainframe. Legacy applications continue to provide value and familiarity while benefiting from the server based architecture and a modern network environment. In many cases, the main fault that users have with legacy systems is the archaic “terminal style” interface; this issue can be quickly and effectively addressed by modernizing the application’s user interface without having to make significant internal programming changes.

TIP Studio OLTP Servers provide an end user interface using standard Internet technologies that only require the use of a standard Web browser on the client device. Since Internet technologies are used, this means that the end user can access the legacy applications using the corporate Intranet or connect in using the global Internet. Of course access to the system is strictly controlled to ensure that only authenticated users with appropriate credentials gain access to the information systems. The user interface can support three levels or styles of user interface as follows:

- **Terminal Emulation Style.** This is used when the legacy application is generating terminal specific codes (i.e. UTS codes and DICE sequences). When this is the case, the resulting display is character oriented and basically looks the same as it would have on a real UTS terminal even though the display is being rendered using a web browser.
- **Screen Format Style.** This is used when the legacy application is using a screen-formatting package, such as DPS. When the application presents results, it delivers data and the name of an externally defined screen format that is used to display that data. The system merges the user data with the screen format to create a complete display. In this mode, the display is rendered in character-format, which will look the same as it would have when running on the mainframe, even though the display is being rendered using a web browser. This visual compatibility is desirable if the impact on end users of phase 1 of the project is to be kept to a minimum.
- **Graphical Form Style.** This is essentially the same as “Screen Format Style” outlined above from the program logic point of view. That is, the programs are still using externally defined screen formats and calling DPS routines to process them. However, in this mode the data is rendered using a graphical form rather than a character oriented screen format. The design and style of the graphical forms used can be as complex or simple as the requirements of the organization dictate.

TIP Studio includes tools that will create an initial graphical form from an existing character oriented screen format. Once this initial graphical form is created, then it can be modified and enhanced using standard forms design tools, such as Microsoft’s Visual Studio. The graphical forms can contain display process logic, typically written in a language such as Visual Basic. This display processing logic can take the data sent from, or to, the legacy application and render it using virtually any graphical technique available. This includes the use of such standard controls as radio buttons and check boxes to richer controls such as tab controls, bitmap displays or even audible controls.

There is literally no limit to how sophisticated the user interface can become using this technique.

TIP Studio Connection Server

TIP Studio Connection Server is a component that provides network transport services designed to ease the transition from the legacy mainframe environment to the new server based systems.

Although the display services provided by the TIP Studio OLTP servers offer significant advantages and a more simplified environment since they are browser-based, it is not always possible to switch all online access to this technique at the time of the cutover from the mainframe to the new server platform. There may not be time to roll out new access procedures to all the end users, or there may be situations where other computers are connecting into the system as terminals and changing procedures on these other computers is just not an option as part of the initial cutover.

To facilitate these issues, IngleNet provides the TIP Studio Connection Server which implements the same terminal connection protocols as those that exist in the mainframe environment. Typically these protocols were provided by front-end communication sub-systems, such as the Unisys DCP. When a client device, be it a PC running a terminal emulation package or a computer connecting in as a terminal for “screen scraping” purposes, connects to TIP Studio Connection Server, there is no difference in protocol from what previously existed using the DCP. This is not to imply that TIP Studio Connection Server provides all the functionality that currently exists

in a DCP system; it does not. However in the specific area of terminal connectivity, TIP Studio Connection Server is a functional replacement for the DCP.

Summary

TIP Studio is software that implements the IngleNet vision. It consists of a number of components designed to meet the requirements of organizations that want to modernize legacy systems by adopting modern technology and use features of the new system to enhance their IT investments.

The use of industry standard Windows and UNIX/Linux based technologies maximizes the return on investment and provides the highest possible value with the lowest risk to your organization. **TIP Studio** software allows you to control the rate of change for your organization, allowing you to set the pace of modernization based on your requirements and resources. Finally, **TIP Studio** lays down a solid foundation for future application development and enhancements by creating a data model designed specifically to meet the needs of your organization.

IngleNet, a proven leader in legacy systems modernization, can preserve the value you have invested in information systems and liberate the potential value of your data through the application of our proven and reliable **OpenARMS** modernization process.

Frequently Asked Questions (FAQs)

1. **Q: Will my operation run faster on the new platform?**
A: Yes. We have experienced many cases where 'order of magnitude' increases in speed have been the result of platform migration projects. Two noted areas of benefit are that programs compile faster and many batch operations are reduced from hours to minutes.
2. **Q: Can you handle Field Data?**
A: Yes. Any data stored using the "field data" character set is translated to ASCII.
3. **Q: Are the APIs that currently exist on the 2200 mainframe available with TIP Studio III?**
A: Yes. At IngleNet, our approach is to implement the core set of functionality that is most used and then to implement less used APIs on an "as needed" basis. When we are engaged in a project, we make an assessment during the initial analysis phase of the project to determine if API's are required in TIP Studio to handle some aspect of the project. If we determine that this is the case, then appropriate time is allocated in the project schedule to complete the work required.
4. **Q: Can I change my SQL schema after I have moved my legacy applications to the new platform?**
A: Yes. The SQL Schema can be changed after the legacy applications and data have been moved to the new platform. In addition to normal database maintenance procedures that would be routinely done when a schema change occurs, there may be an extra step required if the change to the database is something that will ultimately affect legacy applications. When legacy applications are affected by the change then the TIP Database Interface mapping rules also have to be updated to reflect the change made to the SQL database. If however, the change to the database is not going to affect legacy programs such as the addition of a new column or new table that the legacy applications do not require access to, then no additional steps are required beyond those that would routinely be done when the database schema is altered.
5. **Q: Does TIP Studio support clustered operations?**
A: Yes. The Windows version of the TIP OLTP Server and the TIP Connection Server both support clustered operation using the standard Windows Cluster Manager. When used in this manner, the TIP OLTP Server is only active on one node of the cluster. If a failure occurs, the TIP OLTP Server will switch to another node in the cluster and resume operation.
6. **Q: What versions of Windows Servers do you support and recommend for TIP Studio III?**
A: IngleNet recommends a production server environment be run on either:
 - Windows 2000 Advanced Server
 - Windows 2000 Datacenter ServerFor a development server (a developer's PC), IngleNet recommends one of the following:
 - Windows 2000 Server
 - Windows 2000 Professional
 - Windows XP Professional
7. **Q: What versions of UNIX or Linux Servers do you support and recommend for TIP Studio III?**
A: IngleNet support the following versions of UNIX and Linux for use with TIP Studio Version III. The versions quoted are the minimum version required. Later versions of each platform may also be supported. Contact IngleNet for final confirmation. (Note that TIP Connection Server is only available as a Windows Service).
 - Caldera OpenUNIX 8 (as well as SCO UnixWare 7.1)
 - HPUNIX Version 11
 - AIX Version 4.2
 - SUN Solaris (Sparc) Version 2.7
 - Redhat Linux Version 7.2

8. **Q: Do I have to install any client software to use TIP Studio III?**
A: No. All client access to applications running under control of TIP Studio III is performed via a standard web browser such as MS Internet Explorer or Netscape Navigator.
9. **Q: What is the minimum and recommended client configuration for using TIP Studio III?**
A: A standard WinTel PC that will adequately run a version of MS Windows to support the use of MS Internet Explorer Version 6 plus the .NET framework.
10. **Q: How do I run my batch processes in the new environment**
A: Batch runs are processed in much the same way as they currently are processed? The batch runs are initiated from a client terminal as there is no "operator's console" per se on a modern UNIX or Windows server system. The ECL runs streams are converted to scripts as part of the migration project. The script contains the same functionality as the ECL run did. Often the scripts are scheduled to run automatically using standard operating system facilities such as CRON or AT. For more demanding batch processing environments, third party tools can be used to provide the more sophisticated capabilities (job scheduling, performance management, output management etc) required in these kinds of intensive processing situations.
11. **Q: What is a typical system configuration for a 500 user system**
A: These kinds of configuration sizing questions are often asked of Inglenet, however, they are virtually impossible to answer without a great deal of specific information. At the same time, it is important to understand some basics about hardware configuration on open platforms. In the mainframe environment, hardware configuration was a very important planning task since the ramifications of incorrectly sizing a mainframe were quite significant. If the configuration was undersized, then sometime the only solution is to get the next bigger model in the mainframe line. This is an expensive proposition and may involve costly cut-over procedures. If the configuration was oversized, then the organization is paying for computer power that does not require and is not using. With modern server technology, scaling the hardware platform up to meet the needs of the organization is more manageable and less disruptive. It is more important to pick a platform that can be easily scaled to meet the ultimate needs of the organization than it is to select the exact configuration for the job. Plan on adding some hardware resources after the system goes into production. These resources may be memory, CPUs or disk space. The main point is that scaling the hardware is relatively easy and should be thought of as a normal, expected event that often takes place after the system goes into production rather than a failure to accurately predict the exact system requirements.
12. **Q: When should I purchase my production server?**
A: We usually recommend that customers not buy a production server until the majority of the initial migration work is complete. This is simply because the hardware technology has been advancing so rapidly in recent years that delaying a buying decision for 6 or 12 months can mean acquiring a more advanced system for the same or even at a reduced cost. Many customers adopt a practice of having a development/test server and one or more production servers. In this case, we recommend buying the a smaller server initially as the system that will be used to do the modernization work on and once the initial phase of the project is complete, then acquiring the larger production server(s) and continuing to use the migration server as the development/test server.
13. **Q: Does migration improve my security?**
A: The term "improve" is a relative one and depends on the level of security in practice today. Because of their proprietary nature, mainframe systems are regarded as reasonably secure systems and it is a concern when moving to more open technologies that security may suffer. It is reasonable to say that newer, open systems can be as secure as legacy systems.
14. **Q: What happens to Mapper?**
A: If the customer wants to eliminate Mapper as a solution technology, Inglenet can provide services to design and implement a replacement application using current development tool, methodologies and frameworks as prescribed by the customer. Alternatively, Inglenet has developed business partnerships with firms that specialize in Mapper solutions, and where necessary, they are brought in to be part of the transformation team. The Inglenet focus is on DMS data, batch and on-line applications originally written in industry standard languages such as COBOL. Our partners provide the Mapper component. The most typical solution is to move to Mapper-C on either a UNIX or Windows platform.

Q: What do I do with RDMS?

A: RDMS is a relational database available on the Unisys mainframe. Although it is a relational model database, it is not fully compatible with industry standard databases such as Oracle or Microsoft SQL Server. IngleNet's transformation tools will process the current COBOL source code and convert existing RDMS statements into the appropriate statements for the target database.

15. Q: Do you support n-tier software architectures?

A: Yes. This question is not so much “does IngleNet support n-tier architectures”, but rather “do the existing legacy applications support n-tier applications”? The concept of n-tier application refers to the fact that the applications use external services to manage critical segments of the applications. The most common of these external services are database access and presentation services. If the applications are currently using a database such as DMS and presentation services such as DPS then they can be considered n-tier applications and will remain so when they are moved to a new platform. In this case the external services will be different, DMS will be replaced with a relational database like Oracle or SQL Server and the DPS presentation services will be replaced by TIP Studio presentation services which will allow the user to take advantage of modern GUI presentation techniques. *(If the applications do not use these external services, then they are not n-tier, but rather monolithic applications and will remain so when moved to a new platform. Typically these types are applications drive the communications hardware directly with raw UTS codes or DICE sequences. IngleNet will move these application to new platforms, but their operations remain basically as they were on the mainframe with little opportunity for improvement.)*

16. Q: What is the next step after migration/re-platform?

A: This depends a bit on what got done in the initial phase of the project. Often the initial migration phase will not include any attempts to modernize the user interface. In this case, the likely next step would be put some effort into making the user interface more modern and appealing to the end user community. Since this is such a visible part of the application it is a great way to show that progress is being made in the modernization effort. If the user interface issues were already addressed in the initial phase of the project, then the next step is to take a look at the code itself and make some decisions. Ultimately customers want to move away from legacy languages like COBOL and use current development environments. There are a few ways to tackle this issue, including:

- Manually re-write applications as major renovations are required
- Re-code applications on a frequency of use/support basis
- Automatically transform the code using software tools
- Leave the legacy code alone and focus on new applications

The selection of one or more of these techniques is determined by analysing several factors including:

- Frequency of code fixes or renovations
- Stability of staff in supporting legacy languages
- Ability for staff to cope with the introduction of new languages, tools and frameworks en masse
- Resources available to deploy on legacy code transformation projects
- Corporate pressure to build new applications to address new business opportunities

™ TIP Studio is a registered trademark of IngleNet Business Solutions Inc.

™ Windows, .Net, Office, Visual Studio, Visual C#, and Visual C++ are trademarks of Microsoft Corporation.

4310 Sherwoodtowne Blvd., Suite 400, Mississauga, ON, Canada, L4Z 4C4

Phone 905 848 2030 : Fax 905 848 3108

For more information, email info@inglenet.com or visit our website at www.inglenet.com