

Discover, Decide and Reuse / Transform your IT systems

(React to business drivers in an agile manner)

Abstract

Ostia has designed and developed a set of tools focused on the automated 'discovery' of data and application elements. This toolset creates a dependency model which illustrates how data flows through the system. This knowledge can then be used for future development decision making and for ongoing support and maintenance purposes.

Contents

Discover your IT Systems	2
Building new Applications that fully utilise core IT environments	2
Technical Set-up.....	3
Discovery process	3
Portus/Discovery Approach	4
Manual Discovery.....	4
Automated Discovery.....	4
Decide	5
Reuse IT Assets.....	5
Transform.....	5
Implementing your decisions.....	5
Portus/Build	5
Portus/Test.....	6
How can Ostia Help?	7

Discover your IT Systems

The core IT systems of many of our large institutions have been built up in layers over the last 20-plus years by generations of developers using the best-of-breed technologies of the day. The success of these systems enabled businesses to compete and grow. On-going success led to business acquisitions and mergers, each step adding further complexity to already complex core systems. With the advent of the internet, new technology was bolted on to deal with increasing demand for direct customer interaction via web and aggregator sites. To compound the problem the developers who had built the systems over time had left or retired from the organisation thus diluting corporate understanding of precisely what the systems did, how they did it and their interdependencies.

Consequently, many aspects of these systems are now treated by IT as "black boxes" (or "white spaces") that are little understood by today's generation of system administrators, who are, understandably, wary of making changes lest they trigger the sort of high-profile issue that has dogged many an institution in recent years. The favoured approach to this problem has been to add layers of middleware, solving the immediate problem but compounding the long-term challenge. Today's systems have become so complex that not only are they difficult, risky and costly to change but businesses no longer fully understand how they operate from a business perspective.

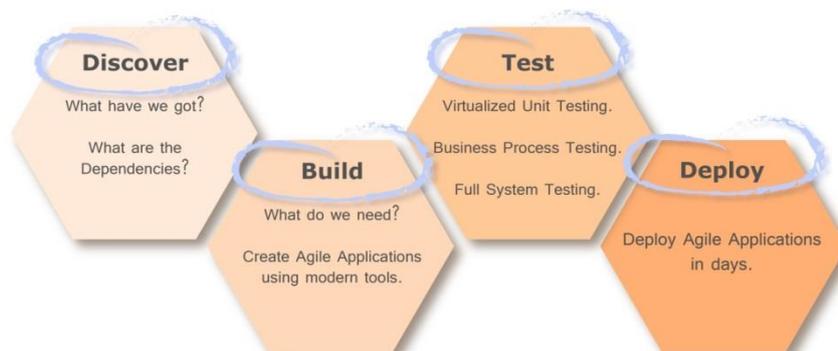
Ostia helps organizations to understand the structure and dependencies of their core transactional systems. Ostia has developed an automated process to document core transactional systems. This creates a dependency model based on the databases and applications running on the system which illustrates how data flows through the system. It also documents what applications are writing to or reading from which tables and columns in which databases.

This knowledge can be used to determine what data or applications can be exposed for the development of new Cloud, Mobile, Web or Social media applications. In addition, this knowledge can be used to determine what data would be useful for analytical or reporting purposes.

Once the knowledge has been collected and decisions about what data or applications are needed, Ostia provide a simple mechanism to create Web Services or APIs around each resource to make them available instantly for the new purpose. This provides an automated, evolutionary process to understand and reuse your core transactional systems quickly and effectively.

Building new Applications that fully utilise core IT environments

The process for building new Cloud, Mobile and social applications around these systems, or analytics projects that use the data on these systems, seems straightforward – It's Discover what we have, Develop what we need, Test and then Integrate it into our processes



If only it were that easy! Often the very exercise of discovery and testing is difficult and complex. This is where Ostia and our Portus technology comes in with Portus/Discover to help you fully understand what you currently have, Portus/Build to create services around what you have and then Portus/Test to enable you to test new applications against the existing environment. The following sections describe these in more detail.

Technical Set-up

Ostia have, over the last 10 years, designed and developed a set of tools focused on the automated 'discovery' of data and application elements. This tool set consists of the following components:

- An Eclipse based control centre and the automated discovery server are installed on Windows or Linux.
- The Portus server component which uses the Apache HTTP Server as its core container. This component is generally installed on the target system. The server component supports a number of different platforms, including z/OS, z/VSE, IS400, UNIX (HP, Solaris, AIX), Linux and Windows. The software is deployed to the target platforms using FTP or SCP. A short platform specific installation is required to finalize the deployment process. The server component will then be running on the core transactional system

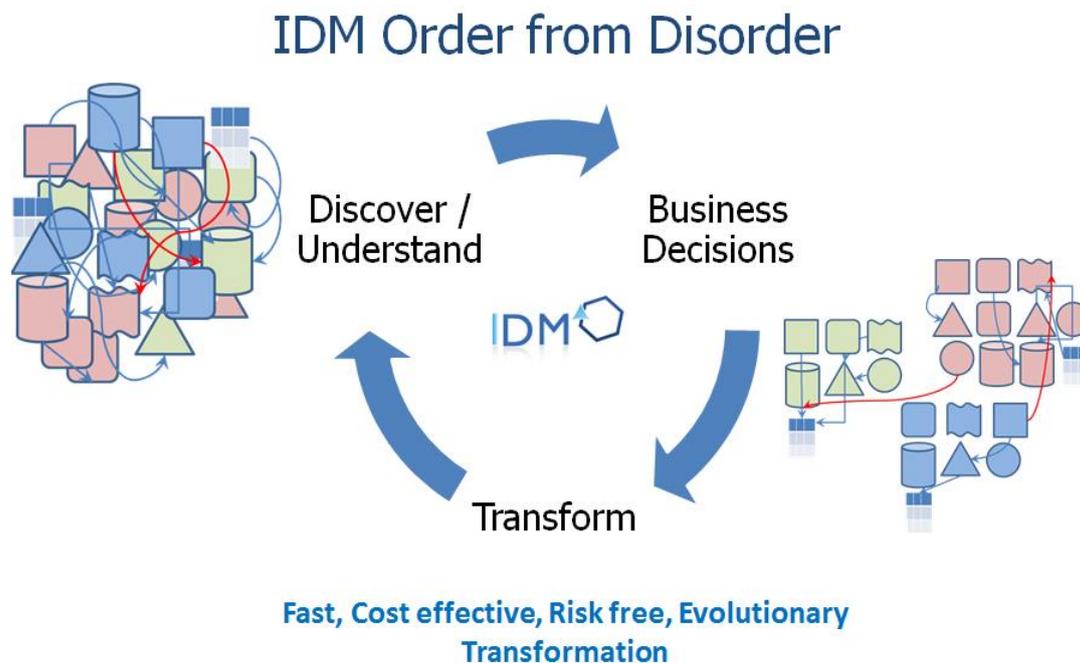
Once these steps have been taken, some small tests are run to ensure that the Eclipse Control Centre and the automated discovery process can connect to the server component. Once connectivity has been established, it is possible to start the discovery process.

Discovery process

There are two potential ways to discover what is on your system. For more simple projects where a limited set of data or applications are in play, the user driven process is likely to be sufficient. Where a larger project or documentation of an entire system is required, the automated process should be used.

Portus/Discovery Approach

Ostia Solution's Portus/Discovery is designed to unravel today's spaghetti and provide a business view of an institution's IT systems without impacting or changing them. Utilizing 'discovery' and 'understand' processes it is able to iteratively analyse the structures underpinning the systems and map the dependencies from a business transaction perspective. This in turn enables the business to reduce risk from an understanding of system inter-dependencies and the ability to perform impact analysis on the systems before updating them.



Manual Discovery

This is managed from the control centre. The control centre uses the Portus server to discover what database tables or files and columns or fields are available. It can also discover what applications are available within the system. In situations where the metadata is available, as in the case of relational databases, the discovery process can complete automatically. Where metadata is not available directly, as in the case of some non-relational databases or application programs, metadata in the form of language sources are used to build the required metadata definitions.

Automated Discovery

This is a more formal and automated approach to documenting core transactional systems. This has the advantage that once set up, it can be repeated regularly, for example, to report on changes that it detects between executions of the automated process. The set up process is as follows:

- The host and port of the Portus server on each machine where data is to be collected are defined to the configuration.
- If only specific sets of data are to be collected on a given machine, this must be defined for each machine. Generally speaking it may be easier to just let it find everything and collect it.
- The collection frequency should be defined so this may be once a week or once a month where the software will run subsequent to the first run and will flag any updates it finds.
- The configuration must include a path which will be searched for source code for any executables that are identified on the core transactional system.

- Where name of the source may be different to that of the executables, rules must be set up to determine how the source for an executable may be established. In the worst case scenarios, a rule may have to be set up to tell the system where to find the source(s) that make up an executable.
- For flat files a similar set of rules must be set up to find structures that will enable the discovery process to understand the flat files.

Once this has been done, the following will occur:

- The collector will build a view of the data model it finds in a local repository.
- It will then process the source of each application it identifies and build a dependency tree showing the application and database column dependencies.
- Once completed a graphical representation of the system dependencies is available.

Decide

Once the discovery process has completed, the business can review their requirements and the data and applications that they need for the business driver they are addressing. This process can ensure that the business and IT can finally discuss something concrete based on the discovery process that has just completed.

Reuse IT Assets

Once the data and applications that are to be reused are identified, Ostia uses a simple point and click configuration based approach to immediately create Web Services/APIs for each resource that is to be accessed. This makes the data or functionality available immediately for use in the new service or application. The new APIs created during this process will re-use the existing authorization and authentication security rules within the organization.

Transform

Where transformation is required from the existing core transactional system, Ostia can provide services to facilitate this process and ensure that it can happen in an evolutionary way.

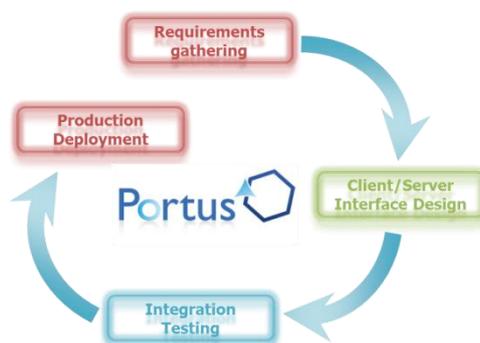
Implementing your decisions

Once you have mapped out your systems and decided on your approach, Ostia offer a pathway to reuse or migration and can help you to achieve is in an evolutionary way. This is done with a combination of a build process to access the resources while also providing a simple, configuration based approach to standing up virtual testing environments.

Portus/Build

Portus/Build is code-free integration. Install Portus and services are created from the business logic or database and deployed to the Portus server within seconds. Once deployed, the Portus service is available for use from multiple languages (e.g. Java, C# .Net, PHP etc.), technologies (e.g. Excel, InfoPath, etc.) and products (e.g. all modern products can invoke a REST and/or a SOAP service). The net result is a much more streamlined approach that reduces cost, time and risk.

Portus/Build Approach



Portus/Test

How Portus/Test can make System Testing less...testing

There was a time when the testing phase meant that you were getting close to project completion. Now it often becomes a project in its own right and one carrying more inherent risk than the development phase did. So exactly how did we manage to achieve that?

In today's austere business climate, IT is expected to deliver high quality, valuable applications earlier and at reduced cost. Often, this must be achieved with globally distributed teams, continually changing requirements and large, complex IT infrastructures.

With the advent of Client / Server in the 90s, application systems acquired a whole array of dependencies on other systems within the organization and even on services outside of the organization. Consequently, in many cases, the only option is to test with a real version of the service despite all of the issues and risks that entails. Such as:

- the service not yet being available preventing testing taking place
- each invocation of the service incurring a charge making testing very expensive
- the service running on an expensive platform, such as an IBM mainframe, also making testing expensive
- the need to create test data in the test environment due to regulatory limitations placed on accessing production data.

To circumvent this, some organizations 'stub' out calls to services for testing using a very simplistic implementation of the service. While this can functionally test code, it has real limitations as it normally requires code changes to invoke the real service which introduces the risk of invalidating the test results. In addition such tests often employ very limited data thus introducing another variant and attendant risk. Consequently such testing is rarely comprehensive and the test data coverage is minimal, giving a real risk of undetected bugs in the released application.

Ostia have developed Portus/Test to meet the challenges encountered in a modern testing environment. You need innovative solutions to succeed in this new environment such as creating virtual service layers that:

- eliminate system constraints
- create stable environments for testing
- increase agility and reduce costs
- shorten development and testing cycles
- enable faster delivery of high quality software.

Portus/Test creates a virtual service layer which replicates the behaviour of your live systems. In doing so, it allows developers and testers to work in a stable, isolated environment, free from the constraints of system unavailability and cross-system dependencies. This minimizes the risk of defects going into production, reduces costly delays and increases efficiency; enabling teams to work in parallel and become agile in responding to change.

However, the real value of Portus/Test comes in its ability to provide intelligent message responses that improve the quality and shorten the length of your testing cycles. Portus/Test allows users to create rich, high quality sets of synthetic test data, as well as mask sensitive data 'in-flight'. Using synthetic data ensures compliance with data protection regulations increases the quality and value of testing cycles and enables users to detect and fix more potentially costly defects earlier in development.

In summary Portus/Test gives you:

- the ability to test your full business process rather than individual services
- a stable testing environment for legacy systems
- high quality, rich sets of synthetic responses that shorten test cycles and improve quality
- a market-leading range of virtualized services
- Dynamic Service Masking that masks data 'in-flight', ensuring compliance and full consistency.

And all without the need for coding!

How can Ostia Help?

Ostia's speciality is addressing the elephant in the room - your core transactional systems. Ostia have many years of experience with these systems and have developed processes and tools to automate the documentation of your core transactional systems. This will facilitate you making informed business decisions around your IT based on the key business drivers you are facing today. Ostia can then help you react to your drivers in an evolutionary way whether you decide to reuse, migrate or replace.

Ultimately Ostia can help you to make your core transactional systems agile and reactive:

- at your pace
- at reduced risk
- with less pressure on your IT function
- at reduced cost

resulting in a more agile and reusable core transactional system that can react quickly to the business drivers of today and of tomorrow.