SOLIX™

Empowering the Data-driven Enterprise

# Cost Savings Opportunities from Decommissioning Inactive Applications

**MARK LEE**

Senior Vice President, Services
Solix Technologies, Inc.
https://www.solix.com

There are few projects in IT easier to quantify cost savings and ROI than projects to decommission inactive applications. Older applications pile up in the corner of the data center as the organization modernizes their portfolio and as acquired companies are transitioned onto the company standard applications. Costs for maintaining inactive applications may include license and support fees for both software and hardware, data center charges, labor costs for application maintenance and backups, and labor costs for application support. The Applications Director for a large manufacturing company described the process of finding savings for his application decommissioning project like this: **"It was as easy as walking through the data center and bending over to pick up hundred-dollar bills"**.

A study done by the Compliance, Governance, and Oversight Council (CGOC)[1] showed that the average annual cost savings for decommissioning inactive applications was $40,000 and that for larger, enterprise class applications, the annual savings could exceed $120,000. With typical application portfolios, large organizations may have hundreds of inactive applications which could yield annual savings of millions of dollars if a cost-effective way to decommission them could be found.

While measuring the potential cost savings for decommissioning inactive applications may be easy, harvesting these savings can be more of a challenge. This challenge comes down to two simple things:

1. the data contained in the inactive applications and
2. the application users that still depend on access to that data to perform their jobs.

An inactive application may contain valuable financial data needed by auditors or it may contain important customer history data needed by the customer support team. Whatever the use case may be, before harvesting the potential savings for decommissioning an inactive application, a company will need to devise a strategy for dealing with this data and the users that depend on it.

The data challenge (discussed below) will remain the same regardless of when the project to decommission the application is started. The end-user challenge, however, will vary significantly depending on this timing. The more time that has elapsed since the application was inactivated, the fewer demands end-users will have for data access. Decommission an application that was inactivated one week ago, and end users will demand nearly the same level of data access that they had when the application was live. Decommission the same application one year later, and the end-user demands for data access will have been reduced significantly.

**"It was as easy as walking through the data center and bending over to pick up hundred-dollar bills"**

[1] CGO "Information Lifecycle Governance Leader Reference Guide"

## The Data Challenge

Data contained in inactive applications may still have significant value to an organization. The data may be needed to comply with industry or corporate policies or data protection and privacy regulations. It may be needed for a variety of business reasons like customer support, product and warranty support, or financial audits. Just discarding this data is not a viable option for many applications.

To harvest the potential cost savings, the data from the inactive application must be moved to a new home. Typically, this new home is some type of archive. The question about this move is how to retain access to the application's data once the application that created it is gone. This challenge is especially difficult for data created by complex, enterprise-class commercial applications (and it is no piece of cake for custom applications either). These applications may have complex data models with tens of thousands of tables and hundreds of thousands of columns (plus associated documents and attachments). How can anyone possibly understand and use the data from a complex data model without the application? This is the ***data model problem***.

To further describe this problem, imagine an enterprise-class financial, manufacturing, distribution, and support application with 50,000 tables and 1 million columns. Further, imagine that this data model was not implemented with any primary or foreign key relationships defined in the model itself (and that all of these relationships were defined in the application code which is going away when the application is decommissioned). Also, imagine that you do not have access to the application source code or technical documentation to begin to understand this data model (or have no interest in trying to do so). Lastly, imagine that this application data model was developed by programmers that loved complexity and used obscure naming conventions for every table and column (as if to make your job even more impossible). Unfortunately, this is the scenario for most major package applications from SAP to Oracle E-Business Suite to PeopleSoft to Siebel to many others.

In the face of this data model problem, what is one to do? A variety of techniques have been tried to varying degrees of success. Some will advocate the use of data modeling tools to analyze the data model and represent it graphically in a way that makes it easier to understand and access. Another technique that is sometimes employed is a code analyzer (assuming access to source code is available). These analyzers can scan the code and discover the hidden primary and foreign key relationships that define key objects in the application. The most common technique is to reverse engineer key screens and reports in the application to determine which tables are being used and how they are related (using log analyzers to view the sql statements being executed by the database). Each of these techniques requires skilled data analysts and archivists. As of the date of this writing, *there is no silver bullet technology that will solve the data model problem without applying a significant amount of technical and application expertise*.

## The End User Challenge

Ask any end-user of their preferred method of archiving data from their inactive, legacy application and you will get the same answer: "No thank you, leave my data alone". Obviously, if you follow this advice, you will not be able to harvest the potential cost savings from decommissioning this inactive application. The conviction of the end-users "No thank you" will vary inversely to the length of time that the application has been inactive. That means that to maximize the potential cost savings, you will have to deal with the most intense end-user resistance. If you wait a few years until the application is forgotten and the intensity of end-user resistance has faded, most of the potential savings will have evaporated. This end-user resistance is a classic change management challenge that was discussed in *Who Moved My Cheese?* [2]. In our situation, we can describe this resistance as *"Who moved my data?"*

Let's look in on a typical scene being played out in a nearby conference room (or Zoom meeting): Ted, the application director, has recently migrated the company's financial and procurement system from a legacy on-premise mainframe application to a modern, cloud-based financial application. Ted wants to decommission the mainframe to avoid a $200,000 maintenance renewal bill coming due at the end of quarter, but he knows that he has to archive all of the accounts payable & receivables invoice data (and other financial records). Sara is the accounting supervisor and the first stop on Ted's end-user approval tour for his decommissioning project.

"So, Sara, what kind of data access do you think you will need from the old financial records when we retire the mainframe?"

Sara replies, "Well, we used one hundred different AP, AR, and GL reports, so I guess I will need one hundred reports".

Ted replies, "but Sara, you have a brand new, cloud-based application to run your department, why do you need all these reports from the old mainframe data?"

Sara replies, "Ted, I'm really busy closing the books, get back to me when you get those reports finished and we can talk..."

In the face of this end-user challenge, what is one to do? One option would be to give in to Sara's demands and re-create every single report from the legacy application to run inside the archive. Another would be to ignore Sara and risk her wrath when the reports go away. Neither of these options is viable. Remember, the reason for decommissioning the legacy application was to harvest cost savings from doing so. If you spend all of these potential savings recreating the application reports inside the archive, what have you accomplished?

Clearly, between the data challenge and the end-user challenge, there will be costs and effort involved to decommission the inactive application and realize the possible saving opportunities. The question is how to meet these challenges in the most cost-effective way to maximize the potential savings?

[2] Spencer Johnson, M.D., Who Moved My Cheese, G.P. Putnam's Son's, 1998

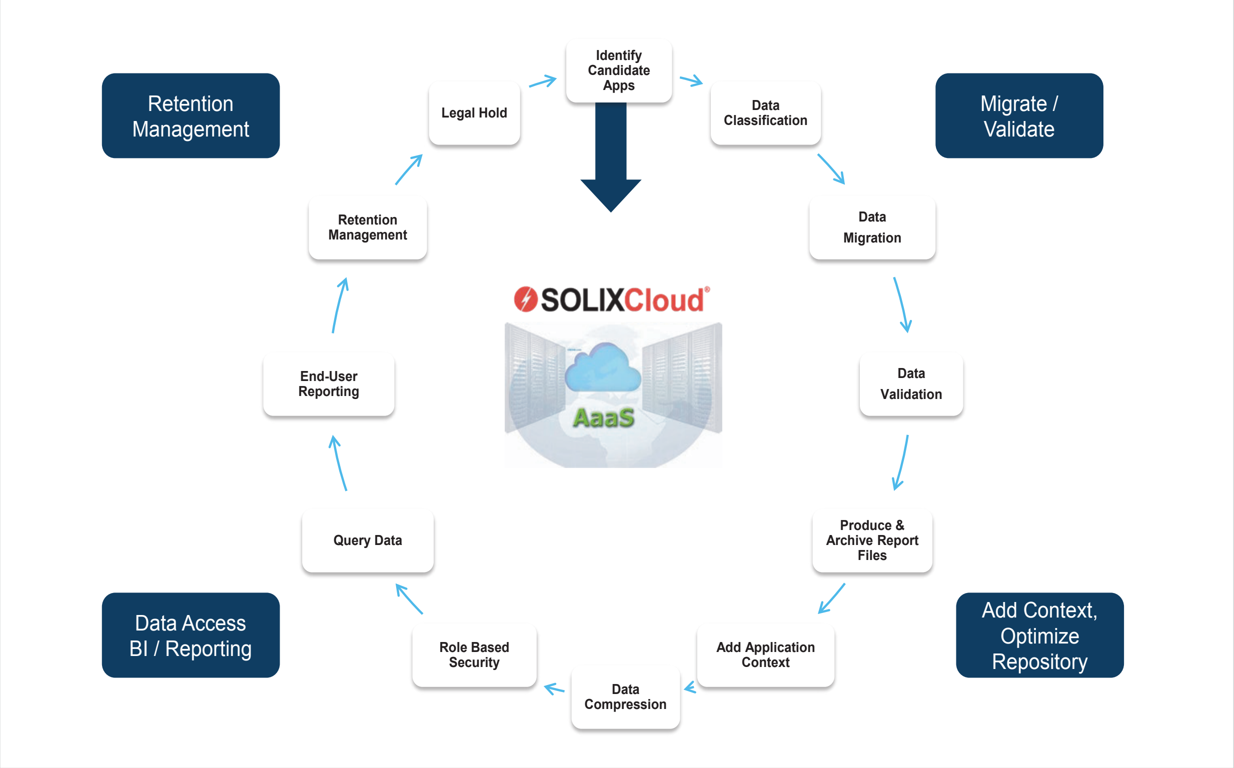
## SOLIXCloud Application Retirement as-a-service

Solix has developed software, a repeatable process and methodology, and a set of factory services specifically designed to solve the application decommissioning data challenge. The SOLIXCloud Application Retirement as-a-service solution consists of three main parts:

1. Solix Common Data Platform (CDP) software-as-a-service in the Microsoft Azure cloud (or private cloud);
2. The Solix Application Retirement Process and Methodology; and
3. Solix Application Retirement Factory Services.

## Solving the Data Challenge

First, it is imperative to efficiently migrate data from the legacy application into the archive. While this may sound simple, it can be quite complex when dealing with large, enterprise-class applications. Solix CDP supports hundreds of legacy databases and sources and can automate the process for migrating data. Once migrated, it is important to validate that the data was moved completely and correctly. Again, Solix CDP automates this process with extensive validation algorithms for every data type. The validation reports produced by these algorithms are key to convincing end users and auditors that, when the time comes, it will be safe to pull the plug on the legacy application.



The crux of the data challenge has to do with ***context***. When the data resided within the legacy application, the context was provided by that application. When the application is removed, that context disappears, leaving the data to stand on its own.

Solix has developed steps in our approach and methodology to preserve as much context as possible to give value back to this data. Many applications contain a data dictionary. Solix CDP can import this data dictionary and add it to the archive repository. The information in this dictionary, while relatively straight forward, can make a huge difference when trying to understand the data model. Imagine a table name "RPXXQ." Unless you were a developer familiar with this application, it is highly unlikely that you could ever know what kind of data was stored in this table. But, if we preserve the user-understandable name for the table "Accounts Payable Invoices", it would be a different story.

Another form of context that can add significant value to the data stored in the archive is static report output. You may have paid millions of dollars for this application and it included important reporting functionality used to run your business. Before decommissioning the application, why not run key reports, save the output to PDF files, and archive these reports alongside the data? Solix CDP provides a way to automate this whole process, called the Virtual Printer (or you can just ingest previously created report files into the archive). Someday soon, an auditor looking in the archive for the old financial reports will thank you.

One of the most important forms of technical context for legacy applications are custom reports that you may have created over the years. These reports contain valuable query code that can be extracted and saved to text files which can be archived along with the data. Preserving these queries in the archive will go a long way towards meeting the end-user challenge as well. By utilizing the Solix CDP SQL Editor, you can run these preserved queries to recreate your custom reports inside the archive.

Because CDP supports both structured data and document data, it is easy to preserve these documents in the archive which helps preserve the context for the structured data. There are other documents that can be preserved, if they exist, which will add to this context including reference manuals, technical documentation, procedure documents, operations documents, training documents, screen shots, printed statements, printed forms, etc.. You may need to keep data from your legacy applications for many years. The people accessing the archive years from now will thank you for preserving all of these documents to help them better understand the data.

As previously mentioned, there is no silver bullet technology to solve the data problem without requiring technical and application expertise. That is why the Solix solution includes a services component: The ***Solix Application Retirement Factory Service***. This service consists of a large group of data analysts, archivists, and developers that specialize in solving the application decommissioning data challenge. The members on this factory team are experts in the techniques previously described like data modeling, code analysis, query log analysis, and reverse engineering. The team also consists of specialists for most of the common, enterprise class applications currently being decommissioned. The benefit for utilizing this factory team is that you don't have to learn to solve the data problem on your own. *Let the Solix archiving experts help you through this hurdle.*

One of the most important benefits of working with the Solix Application Retirement Factory team is gaining access to their extensive library of project accelerators for specific applications.

Over time, this team has decommissioned hundreds of different applications and has built up a library of accelerators to speed up delivery of useful and compelling data access to end users, while keeping costs to a minimum. These accelerators are available for many of the major enterprise applications and consist of pre-defined business object models, reports, and forms. More on business object models later. If you need to decommission one of these applications, the data problem has already been solved for you by utilizing this factory service.

Here is a small sample of the types of applications that have been retired by the factory team:

| CRM | ERP | Financial | Mfg |
|---|---|---|---|
| • KMS<br>• Oracle eBusiness<br>• PeopleSoft<br>• Siebel | • BaaN<br>• Global SSA<br>• Great Plains<br>• JDE<br>• ManMan<br>• Oracle eBusiness<br>• SAP | • Infinium<br>• Mitchell Humphrey<br>• NetSuite OpenAir<br>• Oracle eBusiness<br>• PeopleSoft<br>• Sage FAS100<br>• SAP | • Mfg Pro<br>• Oracle eBusiness<br>• QAD<br>• SAP |

| Procurement | SCM | Talent and HR | Technology |
|---|---|---|---|
| • Ariba<br>• Oracle eBusiness<br>• SAP<br>• SmartBuy | • Oracle eBusiness<br>• SAP | • BIS<br>• Hire Right<br>• Infinium<br>• KBACE<br>• Kenexa<br>• Kronos<br>• Oracle eBus<br>• PeopleSoft<br>• SAP<br>• Training Tracker | • OnBase<br>• SAI (GRC) |

In our view, the key to solving the data problem in an application decommission project is to understand the business object models used by the application. For example, in an accounts receivable application, the key object model is the invoice. In a general ledger application, key objects are journals and budgets. In a customer service application, a key object is the service order. In most applications the number of objects is typically a fraction of the number of reports / screens. If we can focus on defining these key objects rather than on replicating reports, we can cut the work on the retirement project down to size.

How can this be accomplished? Solix CDP introduced the concept of the ***Enterprise Business Record (EBR)*** to accomplish this goal. The EBR is a model of the entire business object (including both structured and unstructured data) stored in a denormalized, flattened structure. By itself, a model is just a model. What good is it? In CDP, the EBR comes to life through the power of ***search***. Since Dr. E.F. Codd invented the relational database, structured query language (SQL) has been the lingua franca for database access. In the early days of SQL, it was a huge leap forward in usability and allowed technically proficient users and IT professionals to access data without having to write programs in lower level languages. With the advent of modern reporting and analytics tools, users no longer had to write any SQL at all because the tools did that for them. They just had to have an understanding of the data model.   Which brings us back to the original data model problem. The EBR with search changes this paradigm. Now, a business user can access their data without having to know anything about SQL or about the underlying data model.

They only need to know how to use a text search tool. The EBR with Search enables the concept of ***self-service, end-user data access***.

With a modest investment to define the application's key business objects as EBRs, and the power of search enabled by SOLIXCloud, an organization can meet the end-user data access requirements and keeps costs low to maximize the savings that can be harvested and be returned to the IT budget for use on more strategic initiatives.

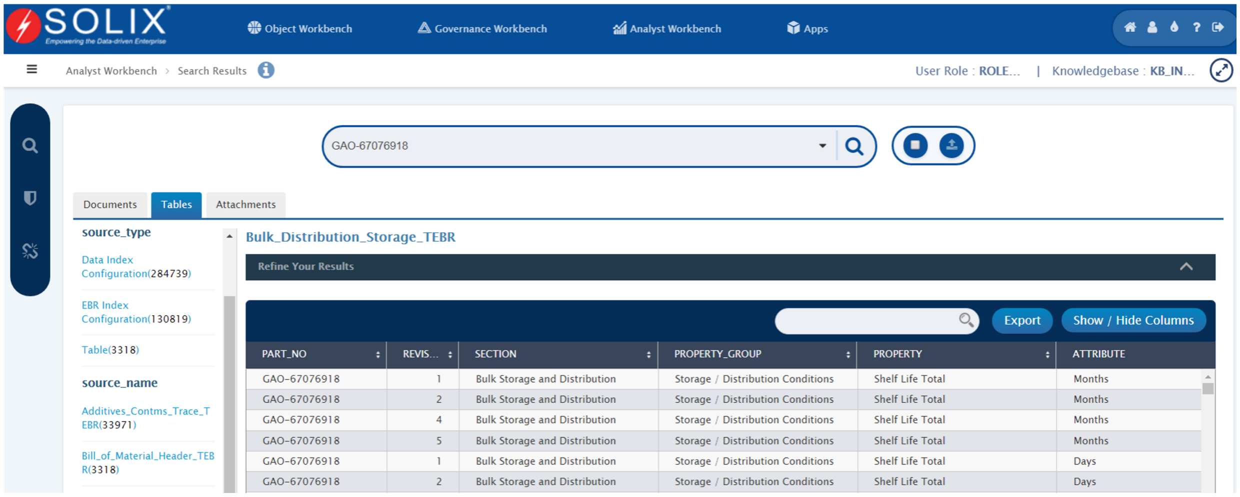
## Solving the End User Challenge

The reluctance of application end users to allow their data to be moved to an archive should not be a surprise. Every end user organization has their own mission and challenges and asking them to take some of their valuable time to learn a new way to access data will need to be justified. For many end users, the answer will be to either leave the application in place or to replicate all of the screens and reports in the application before decommissioning it. As we have seen, neither of these options are viable if we wish to harvest any cost savings.

The answer for many organizations is the adoption of self-service, end-user data access powered by EBRs and search. As we have seen, one EBR representing a key business object in the application can replace many screens and reports. Coupled with the project accelerators provided by the Solix Application Retirement Factory team, which include out-of-the-box EBR definitions for most enterprise class commercial applications, you have the start to a very cost-effective solution to this challenge.

Solix CDP enables self-service, end-user data access through a workbench of tools for ad hoc query, structured reports and text search. Role-based security enforces access control in Solix CDP, meaning users can only retrieve data that they have been authorized to see. Once an end-user retrieves their search result, the data can be easily loaded into Excel where the user can slice and dice, sort, and format the data as they see fit. This capability makes it possible for a single EBR to replace multiple SQL based reports and forms and more importantly, eliminates the user from understanding the complexities behind the data.

Here is a simple search example from a decommissioned PeopleSoft application:



GAO-67076918

source_type

Data Index Configuration(284739)

EBR Index Configuration(130819)

Table(3318)

source_name

Additives_Contms_Trace_TEBR(33971)

Bill_of_Material_Header_TEBR(3318)

Bulk_Distribution_Storage_TEBR

| PART_NO | REVIS... | SECTION | PROPERTY_GROUP | PROPERTY | ATTRIBUTE |
|---|---|---|---|---|---|
| GAO-67076918 | 1 | Bulk Storage and Distribution | Storage / Distribution Conditions | Shelf Life Total | Months |
| GAO-67076918 | 2 | Bulk Storage and Distribution | Storage / Distribution Conditions | Shelf Life Total | Months |
| GAO-67076918 | 4 | Bulk Storage and Distribution | Storage / Distribution Conditions | Shelf Life Total | Months |
| GAO-67076918 | 5 | Bulk Storage and Distribution | Storage / Distribution Conditions | Shelf Life Total | Months |
| GAO-67076918 | 1 | Bulk Storage and Distribution | Storage / Distribution Conditions | Shelf Life Total | Days |
| GAO-67076918 | 2 | Bulk Storage and Distribution | Storage / Distribution Conditions | Shelf Life Total | Days |

Based on our experience, EBRs address many of the use cases through self-service. However, Solix CDP also provides the flexibility of other data access options including online forms, fixed format reports, ad-hoc SQL reports, use of existing SQL code, and access to data using other reporting and analytics tools.

For use cases where the data must be retrieved in real time and frequently, Solix CDP provides a forms functionality to replicate an application inquiry screen. This might be useful for a customer support team that needs to look up customer history or product warranty information from the archive while the customer is on the phone.

For situations where a fixed format report is needed, Solix CDP provides a graphical report writer. This can be useful when the same information needs to be retrieved frequently and delivered in a fixed format to the requestor. A fixed format report might be useful for an accounting team needing to provide an account reconciliation report to an outside auditor in an Excel workbook. Or, an HR team might need to produce a timesheet report for an employee or department and deliver the data in a PDF format. The CDP report writer would be an appropriate tool for these use cases.

There may be situations where ad-hoc queries are required as well as searches, forms, or reports. For this use case, Solix CDP provides the SQL Editor functionality. Typically used by an IT professional, the SQL Editor allows for direct entry and execution of a SQL statement (assuming the user has been granted access to the requested tables) as well as execution of sql statements stored in text files. As previously noted, a best practice recommended by the Solix Application Retirement Factory team is to preserve custom SQL report code in the archive for future use. Not only can this code provide valuable insight into the data model, but in many cases, it can be executed directly against the tables now stored in the archive. Solix CDP uses an ANSI SQL compatible query engine, so the custom SQL code is likely usable as-is and in rare cases it may need to be tweaked to work in this environment.  If a large number of custom reports existed for the custom application and it is possible to harvest the SQL code from these reports, re-creating them using the SQL editor can be a very cost-effective way to meet the end user data access requirements. These queries can also be saved for future reuse.

The final data access functionality option available in Solix CDP is the CDP API. This API makes it possible for end users to access data in the archive using other data reporting and analytics tools. The CDP API uses a standards-based REST technology, making it compatible with most modern tools.

Now, what is the best way to meet the end user resistance challenge? Solix recommends minimizing the amount of effort spent up front building data access. The most common mistake made by organizations is to assume that all of the reports that were needed when the application was in production running the business will be needed after the application has been decommissioned. All too often, organizations will build expensive reports up front and find that they are rarely or never used. If it seems that this recommendation to minimize reporting seems to be in direct conflict with the end-user resistance challenge, it absolutely is. So, what is to be done? A balance must be struck between the end-user reporting demands and the cost of the project (which will eat up potential cost savings).

That balance can usually be found with the combination of self-service end-user data access powered by EBRs and search, and the use of saved custom sql reports run via the CDP SQL Editor. No two organizations or applications are the same but using these techniques will typically lead to an acceptable compromise between IT and their end users.

## Putting it all together

A successful application decommissioning project must be delivered for a price that is less than the potential cost savings. There are some exceptions to this rule such as the need to bring legacy data into compliance with new data protection and privacy laws, or the urgent need to move data off of an unstable and unsupported hardware platform. But, for most projects, if the price of the project is greater than the potential savings, the project will not be economically viable. The successful project must also solve the data problem and the end user resistance problem within these constraints.

SOLIXCloud Application Retirement as-a-service can meet these challenges. SOLIXCloud can solve the data problem with the help of data experts on the factory team and their library of project accelerators. SOLIXCloud also solves end-user resistance problems with the use of self-service end user data access powered by EBRs and search. Lastly, SOLIXCloud can deliver your application decommissioning project for the lowest possible cost per application yielding the greatest savings for your organization.

Let’s look back in on Ted and Sara who have just attended a demonstration of SOLIXCloud Application Retirement as a Service.

“So, Sara, what do you think of the EBR concept?”

Sara answers “Does this mean I can search for my invoices, journals, and budgets and get the data into a spreadsheet on my own?”

Ted replies “Absolutely”.

“And can I sort and format the data any way I want?”

“Absolutely” says Ted.

“And does this mean I won’t have to submit report requests to your team and wait two weeks to get my data?” Sara asks.

“Absolutely” says Ted.

“Then count me in, you have my support” says Sara.

“Great, that means, we will save a huge amount of money by decommissioning this application to the Solix Cloud”, says Ted.

“Awesome” says Sara, “Here is my enhancement list for the new application. Please use those savings to get these done. By the way, we need the enhancements done by next quarter.”

And, so ends another chapter in the never-ending circle of application lifecycle.

SOLIX™
Empowering the Data-driven Enterprise

## Contact Us

For more information contact us at:

**Solix Technologies, Inc.**
4701 Patrick Henry Dr., Bldg 20
Santa Clara, CA 95054

**Toll Free:** +1.888.GO.SOLIX (+1.888.467.6549)
**Telephone:** +1.408.654.6400
**Fax:** +1.408.562.0048
**Email:** info@solix.com
**URL:** http://www.solix.com

Solix Technologies, Inc. is a leading cloud data management application solution provider that empowers data-driven enterprises with solutions for digital transformation including Enterprise Archiving, Enterprise Data Lake, Consumer Data Privacy and Enterprise Content Services. SOLIX-Cloud Common Data Platform provides an Information Lifecycle Management (ILM) framework to provide compliance and data governance for both current and historical data. Solix Technologies, Inc. is headquartered in Santa Clara, California and operates worldwide through direct sales and an established network of value added resellers (VARs) and systems integrators.