The explosion of data in corporations of all sizes is not a new phenomenon, and the growth of stored data shows no signs of abating. Companies are struggling to keep their heads above water as a flood of data engulfs organizations that increasingly rely on IT applications to support their businesses. More specifically, the inability to effectively manage data growth threatens to prevent firms from building and maintaining the applications that, now more than ever, determine how they compete and succeed in their industry.

With this in mind, the time has come for all companies to explore Data as a Service (DaaS) as an option for improving agility and increasing infrastructure utilization. DaaS is a fairly new concept to IT professionals, many of whom are accustomed to storing application data on expensive and redundant infrastructure, and leveraging complicated administrative processes to deliver that data. DaaS can play a strong role in accelerating critical application development (AppDev) projects through self-service provisioning and management of application data in non-production environments. Allowing users, such as developers and testers, to provision and manage their own application data removes key process bottlenecks that slow application projects. After many years of lacking viable options for self-service data management, most companies have understandably come to accept that the status quo is unavoidable. DaaS, however, challenges existing AppDev processes by proving that there is now a new, better alternative to the status quo.

Many AppDev teams are hamstrung by the glacial pace of deploying data in Dev, Test or other environments used to verify application behavior and data integrity. In fact, many AppDev teams face a high hurdle in deploying and maintaining application data in their downstream environments. Requests to refresh or reset data used for application projects can take many hours or even days to complete. These delays can be a result of administrative workloads or technical constraints, such as the time it takes to restore a large amount of data from backup. They can also be caused by the less robust, lower performance storage arrays that typically support Dev, Test or other non-production data environments.

What is DaaS?

DaaS solutions first capture data from key enterprise applications. Next, they control data across the application lifecycle, with functionality to version data as it changes and manage user access. Finally, DaaS solutions create virtual copies that can be used by developers, testers or business analysts. DaaS virtual copies can be created in minutes and take up a fraction of the infrastructure of regular physical data copies, without requiring antiquated truncating or sub-setting methodologies.

DaaS application data may reside in corporate data centers, hosted data centers, or in the cloud. Depending on customer requirements, DaaS solutions can support a wide range of data sources, using cloud or on-premises environments, and on either physical or virtual data platforms. Storing application data in the cloud offers more resiliency and more efficient infrastructure utilization, though there are many reasons why a company might choose to store their application data locally, including faster performance or the need for specific security protections that are better served through local data platforms.
What problems do DaaS solutions solve?

In addition to solving physical and resource issues associated with local data, the right DaaS solution radically changes AppDev workflows and streamlines data provisioning. While IT administrators are typically tasked with managing and deploying data when and where it is needed, application development teams are usually not the first priority for IT operations (IT ops) teams. As a result, turnaround time on data deployments for non-production environments can be slow. Delays in the deployment of data can add days or even weeks to application development and testing schedules. With self-service capabilities, DaaS solutions have the potential to fundamentally change the way companies develop and test applications. The ability for end users to deploy — as well as refresh, share, or rollback — their own data, puts developers and testers back in control of critical application projects.

What parts of an organization can benefit from DaaS?

There will be many beneficiaries to a DaaS approach to data deployment and management, including lines of business users, application development teams, and IT ops teams. Each group has distinct data requirements that can be positively impacted by a DaaS strategy.

Business users

The key benefits of DaaS include faster time-to-market for revenue-generating applications, better security and compliance during the development process, and higher quality applications that mean more uptime for business-critical apps. DaaS also accelerates reporting and business intelligence functions. While many business users are accustomed to spending days or even weeks generating quarterly and annual financial reports, DaaS offers the potential for critical fiscal data to be offloaded from production data sources onto virtual datasets. Financial reporting can then proceed without impacting the production data source. Similarly, business intelligence and analysis can be performed using a virtual database that stays synchronized with a production data source.

Application development teams

In many cases, application development teams stand to gain the most from implementation of a DaaS solution. Faster deployment of application data drives more development and test cycles, which means more features and releases in a shorter amount of time. Developers get access to fresh production data in minutes, and no longer need to hoard application environments to prevent other teams from consuming coveted resources. DaaS also offers access to higher quality test data sets by making environment refreshes automatic, and offering full sets of real data rather than synthetic or sub-setted data. Better test data allows AppDev teams to deliver higher quality applications that need less support.

Many IT shops share test data among multiple development teams, however this can lead to inefficient allocation of access to shared testing resources. For example, one team might only be able to test code within a narrow window during which it is granted exclusive access to the data. In contrast, DaaS offers an efficient solution to key test data management challenges. By giving every team timely access to high-quality data in their own virtual environments, bugs can be captured early when they are less expensive to fix.
IT Operations

IT ops teams have the potential to see a significant positive impact on their activities due to the advantages of a DaaS strategy. While application teams always want to move faster, IT ops teams have requirements around efficiency, security, and controlling resources. Key DaaS capabilities include the masking of personally identifiable information (PII) when production data is deployed in non-production environments. DaaS solutions also offer the capability to replicate data quickly and efficiently within data centers, between data centers, between data centers and clouds, and between cloud platforms (including dissimilar cloud platforms). Once again, the self-service potential of DaaS provides a quantum leap forward in streamlining the management of data. Another key DaaS benefit for IT ops is the ability to deploy as many as 20 application data environments for the price and storage requirements of a single traditional environment running on dedicated hardware. Lastly, DaaS can be an excellent backup and recovery tool for application data, offering highly granular RPOs and quick RTOs.

Summary

DaaS has the potential to transform the way that data is managed in the modern, data-driven organization. Perhaps the greatest benefit to business users, application teams, and IT ops, is the elimination of days or weeks of schedule from data-centric activities including application testing, financial analysis, and business intelligence reporting. There are also additional benefits to DaaS beyond simply speeding up these processes through self-service and fast data delivery: DaaS can also increase security, support faster app release cycles, and ensure compliance with government regulations and compliance guidelines. DaaS, as a technology, has matured enough to make a compelling argument for moving towards a strategy that embraces virtual data for non-production and business line data. Organizations have the ability to dramatically increase agility while avoiding cost by switching to a DaaS-based model for self-service data delivery and management.
Common DaaS Key Performance Indicators

Based on our research, here is a list of important metrics for you to consider when evaluating DaaS solutions against your current non-production data infrastructure:

- What percentage of requests for data set refreshes are met within a reasonable amount of time?
- Do application teams have all the environments requested for Dev, Test, QA, sandbox, and training?
- What percentage of your current non-production environments need masked data to remain compliant with privacy laws, but are not masked because it takes too long?
- What percentage of non-production environments are sub-setted to reduce cost?
- What percentage of data requests are self-serviced by AppDev users without DBA intervention?
- What percentage of data management requests require involvement of privileged IT ops users such as system administrators, DBAs, or storage administrators?

In addition, here are important questions to consider when evaluating one DaaS solution against another:

- Does the DaaS solution support recovery of applications and data to any point in time without manual effort or administrator involvement? Or does the product rely on scripts and workflows?
- To what extent does the solution support data versioning, branching, sharing, and other collaboration capabilities that support common DevOps patterns?
- How does the solution mask data? Can the solution apply masking while maintaining referential integrity across multiple data sources?
- Can you mask a data set once, and then make multiple copies of that masked “parent”? Do the “children” inherit changes from the parent?
- How intuitive is the user interface? To what extent can the user control application data via self-service?
- Does the solution support platform migration (e.g. UNIX to Linux)?
- Can the solution group together multiple data sources and application instances? Can they be synchronized to the same point in time for integration testing purposes?
- Does the solution have a physical limit on the total number of individual copies you maintain?
Mapping DAAS requirements to requisite features and criteria

This hands-on product validation examines Delphix, a DaaS product that streamlines the process of data deployment and maintenance in non-production and other specialized environments. DaaS solutions, such as Delphix, have the potential to change the entire data provisioning process such that application developers, testers, and even business users can directly deploy application data in non-production environments via self-service. AppDev teams no longer have to wait for IT administrators to provision and deploy non-production data suitable for application development, verification, and testing. Rather, they can now perform those data management tasks themselves in a matter of minutes. While the Delphix value proposition includes a variety of benefits, the self-service capabilities of Delphix, in particular, offer a profound opportunity to reengineer a process that is manual and slow, into a streamlined process that lets AppDev teams concentrate on what they do best—develop applications.

Business Line: Requirements, Features/Criteria and Our Take

The table on the next page compares five common business line DaaS requirements, as well as the features and criteria we used for our validation testing and Our Take on the validation results of Delphix for these requirements. See Figure 1 for a screenshot of the Delphix admin console.

---

Figure 1: The Delphix admin console uses a timeline slider at the bottom to choose a recovery point. Once you click on a point-in-time, a card pops up at the top to show you the details for that time.
### Requirements vs. Features, Criteria

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Features, Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business agility: Faster time-to-market, more application release cycles</td>
<td>Self-service data features to enable faster development and more test cycles</td>
</tr>
<tr>
<td>Governance model: Proper controls, privileges across app and IT teams</td>
<td>Self-service data management for full application stacks: separate APIs, web services, interfaces, role-based access controls and privileges per team</td>
</tr>
<tr>
<td>Compliance: Close data security holes, reduce privileged user access risk</td>
<td>Automated delivery: Reduce admin user data handling and surface area of risk</td>
</tr>
<tr>
<td></td>
<td>Integrated masking: Reduce access to sensitive data (dev, QA and analysts)</td>
</tr>
<tr>
<td></td>
<td>Independent data record for fast recovery from data vandalism</td>
</tr>
<tr>
<td>Flexible data collection, access and reporting to enable decisions</td>
<td>Constant automated load of data from source; offload batch jobs to near real-time environment</td>
</tr>
<tr>
<td>Environment consolidation, e.g., reduce X apps down to Y apps</td>
<td>Easily capture non-production environments, quickly provision them to prove on-demand access and then release underutilized resources</td>
</tr>
<tr>
<td></td>
<td>Reduce server/software footprint and storage through de-duplication and higher utilization</td>
</tr>
</tbody>
</table>

### Our Take: Delphix and Business Line Requirements

Delphix provides faster time-to-market for application development and testing due to self-service data provisioning capabilities. Quicker provisioning of application data leads to faster and more frequent application release cycles. We validated that Delphix allows data users to leverage the self-service portal to quickly and easily create, refresh, replicate, mask, and bookmark data. Users no longer need to wait on a change ticket or a DBA in order to deploy or maintain data in non-production environments.

Through our hands-on evaluation of Delphix, we tested comprehensive internal security policies and access controls based on role or team membership that should meet most governance requirements with ease.

The Delphix data masking process ensures that developers and application testers do not have access to personally-identifiable information (PII), as required by government regulations. Because Delphix maintains its virtual data independent from the source data (typically a production database), Delphix can be leveraged to provide timely recovery of lost data due to accidental or malicious deletion. We verified the ease with which Delphix admins and users can restore, refresh, bookmark, and reset virtual databases in both the user and admin console.

We configured and modified various automation capabilities within Delphix and confirmed that configuring automated data loads on a recurring or ad-hoc basis is easy and intuitive. The automation within Delphix makes it easy to keep virtual data up-to-date on any recurring schedule, as required.
How Delphix Works

The Delphix DaaS platform brings virtualization to application data. Much like a hypervisor abstracts compute resources to virtual machines, Delphix abstracts data from storage and other hardware to create virtual copies of data. Delphix accomplishes this through four key steps:

1. Delphix creates an initial, full copy of application data using native backup protocols. For example, it uses RMAN for application data residing on Oracle. Execution time depends on the size of the data source, though this step only occurs once.

2. Next, Delphix stays in sync with the production data source by collecting only change data. This allows the platform to minimize load and impact on the production source. Moreover, captured data is processed into a comprehensive record of data changes over time, allowing the platform to serve as a “time machine” for application data. All captured data is compressed and filtered of temporary or empty blocks to ensure that data is stored efficiently.

3. Users can now create fully read/write-capable virtual copies of application data, a process that typically takes a matter of minutes. Delphix intelligently shares the data blocks that represent the state of an application, as of any point in time. Rather than duplicating or moving data, Delphix identifies the set of data blocks associated with a specific point in time, and shares those blocks with the target. Multiple views of the data blocks – representing multiple virtual copies – can be created and utilized independently. Changes introduced into one virtual copy are tracked separately, so as to not conflict with the state of other virtual copies.

4. Finally, Delphix enables self-service data management by automating all the steps in data delivery including remote mounting of data, resetting initialization parameters, and restarting target environments. In addition, Delphix includes a user interface called Jet Stream that allows end users to directly provision and manipulate data. With Jet Stream, developers or testers can refresh data, rollback data to an earlier point in time, or share a copy of data, all via self-service.

Figure 2: Delphix is deployed as a software layer that resides in between enterprise storage and applications, serving as a virtual data provider to non-production environments.
Application Development: Requirements, Features/Criteria and Our Take

The following table compares five common application development DaaS requirements, as well as the features and criteria we used for our validation testing and the Delphix validation results for each of those requirements. Figure 3 shows the bookmark feature of Delphix, part of the self-service console.

![Delphix Jet Stream interface](image)

Figure 3: The Delphix Jet Stream interface allows end users to navigate application data history, while also enabling them to perform their own branching, data refreshes, restores, and resets at the click of a mouse. The following use cases are illustrative of how an end user might utilize Jet Stream functionality:

- A QA user who finds a bug instantly bookmarks and shares data with a developer to enable rapid troubleshooting against a consistent data set.
- Trainers can instantly reset environments between successive classes.
- BI analysts can refresh multiple reporting copies from their respective sources in minutes.
- A performance engineer branches one data set into three copies with different indexing schemes to test and compare response times.
- A business analyst creates a Jet Stream “container” that groups the latest financial transaction data, data from the last week, and data from the last quarter. The full container can be refreshed, reset or rolled back to an earlier point in time.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Features, Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-service data enablement</td>
<td>User interface suitable for end users (developers and testers)</td>
</tr>
<tr>
<td></td>
<td>Separate APIs (admin vs users), interfaces, role-based access controls</td>
</tr>
<tr>
<td>10x faster dev, test cycles within environments</td>
<td>Data controls including self-service refresh, reset, bookmark, branch, integrate</td>
</tr>
<tr>
<td>2x faster test-fix cycles: Capture, pass</td>
<td>Data controls including self-service bookmark, branch, share, integrate</td>
</tr>
<tr>
<td>environment data with bug reports from QA to dev</td>
<td></td>
</tr>
<tr>
<td>Testing across integrated applications</td>
<td>Integrated log shipping, management and ability to dial data to specific transaction boundaries or down to the second</td>
</tr>
<tr>
<td>(enterprise release testing, management)</td>
<td>Containerization: Ability to logically group elements of an app stack into containers so all elements (app binaries, configuration files, DB) can be provisioned in sync</td>
</tr>
<tr>
<td>Higher data quality in IT environments</td>
<td>Self-service refresh</td>
</tr>
<tr>
<td>(data age, size)</td>
<td>Lowest cost of full dataset as opposed to subsets</td>
</tr>
<tr>
<td></td>
<td>Automatic, integrated masking</td>
</tr>
</tbody>
</table>
Our Take: Delphix and Application Development Requirements

Our hands-on validation confirms that self-service capabilities are tightly controlled via internal security policies and roles. Intrinsic security policies and separate APIs are readily available in Delphix for both admin and regular user roles.

In our testing, Delphix offers almost instantaneous refreshing of data, including complete resets, the ability to go back to any specific point in time, the ability to bookmark a point in time for easy future reference, and the ability to branch data from a specific point to spawn off a data copy branch for other uses. Delphix makes it easy and fast for developers to recreate a bug or other issue discovered in Test using bookmarks, branches, and shares.

The only way Delphix can offer down-to-the-second granularity for database restores is by actively managing the log files associated with virtual data. Delphix automatically manages and gathers database logs that provide that data granularity. We confirmed that we could reset or restore from any point in time on the Delphix timeline user console. This granularity through log file parsing is also how Delphix can retain 30 days of data into the same space occupied by just one day’s worth of physical data.

We validated that Delphix is truly an application-aware DaaS solution in that users can bundle related files and data together to easily transport and deploy entire applications with all files in-sync. These application bundles can be used to seamlessly distribute identical applications with the underlying database and all dependencies intact. We also confirmed that allowing users to refresh their own data can save IT organizations time and money, while accelerating application development and release cycles.

Because Delphix offers customers higher quality test data, Delphix contributes to higher quality applications. Higher quality data means that bugs can be caught and fixed earlier in the development cycle, saving time and money. The ease and speed with which Delphix can propagate full datasets provides timely, cost-effective non-production data environments.

We tested the data masking capability within Delphix and confirm that Delphix offers integrated, automated data that ensures all governance and compliance regulations are strictly observed.
IT Operations: Requirements, Features/Criteria and Delphix Results

The following table compares five common application development DaaS requirements, as well as the features and criteria we used for our validation testing and the Delphix validation results for each of those requirements.

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Features, Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud ready: New products, architectures must work across physical servers,</td>
<td>Any-cloud architecture: run in physical servers, private clouds, public clouds</td>
</tr>
<tr>
<td>private clouds, public clouds (e.g. vCloud Air, AWS)</td>
<td>Replicate across architectures to eliminate cloud lock-in</td>
</tr>
<tr>
<td>IT OpEx: Reduce administrative complexity, expense for projects</td>
<td>Centralized management and self-service: separate APIs, web services, interfaces,</td>
</tr>
<tr>
<td></td>
<td>role based access controls, privileges per team</td>
</tr>
<tr>
<td>IT CapEx: Improve utilization, reduce server, software, storage and backup</td>
<td>Elimination of administrative touchpoints</td>
</tr>
<tr>
<td>costs</td>
<td>Server, software reduction via fast self-service cycles</td>
</tr>
<tr>
<td>Transform UNIX data into Linux data</td>
<td>Significant storage reduction</td>
</tr>
<tr>
<td>Improve, optimize data protection</td>
<td>Effective backup elimination</td>
</tr>
<tr>
<td></td>
<td>Automatic UNIX to Linux conversion</td>
</tr>
<tr>
<td></td>
<td>Integrated log shipping and management</td>
</tr>
<tr>
<td></td>
<td>Strong RTO, RPO</td>
</tr>
<tr>
<td></td>
<td>Full data granularity</td>
</tr>
</tbody>
</table>

Figure 4: Creating and provisioning a virtual database in the Delphix self-service portal. We were able to create a new virtual database in less than a minute during our tests.
Our Take: Delphix and IT Operations Requirements

Our testing confirms that Delphix offers separate APIs and user interfaces for admin and user roles. Policy-based access control and rights are used to reduce administrative overhead and we find these features to be effective. The self-service capabilities of Delphix make virtualized data management a simple task. The user interfaces are intuitive and follow a consistent look and feel.

Delphix reduces storage requirements via de-duping and compression where appropriate. In cases where data has already been compressed, there is no additional benefit (and a significant performance penalty) to attempting compression. We confirm that Delphix contains compression algorithms that prevent wasting cycles on attempting to reduce the footprint of data that has already been compressed.

Delphix provides an intrinsic backup capability for non-production environments, which are often not backed-up. Unprotected non-production environments represent a large risk. Should a disaster or other large-scale data loss occur, AppDev teams could be idle for days without backups from which to restore.

We validated that Delphix can automatically transform UNIX data into Linux data during efforts to upgrade legacy server platforms to current technology.

We verified that Delphix utilizes log shipping capabilities to enable highly granular data bookmarks and resets, down to the second. Due to this granularity, Delphix supports much shorter RTOs and deeper RPO goals than typical cloud-based data protection schemes. Though not designed to be a comprehensive disaster recovery solution, Delphix does include enough automation and replication capabilities to provide backups of non-production data that might have not been protected in traditional IT shops.
Summary

Delphix aims to revolutionize the way that companies deploy and manage non-production (and other) data, and our hands-on validation indicates that they have succeeded with that goal. We believe that Delphix provides unique time saving capabilities that can be leveraged almost immediately after installing the Delphix engine. Delphix can reduce or eliminate the traditional delay required to deploy application data in non-production environments. As a result, Delphix can significantly accelerate the application development and testing process. Delphix touts its ability to store 30 days of virtual data in the same amount of disk space as a single day of full physical data. Companies can leverage this savings to reduce the amount of storage for some applications, or simply enjoy the additional data restore points far longer than is possible with traditional physical data stores.

Compliance and governance is now a top priority of companies who fall under HIPAA, FINRA, PCI, and other regulations. Although other companies market data masking tools, those tools are notoriously slow and hard to use, thereby adding yet another impediment to non-production data delivery. Delphix offers automated masking of PII in virtual databases and can mask data in a matter of minutes (depending on the size of the data source), delivering masked copies of production data with different names, different dates of birth, with fictional social security numbers and addresses. Delphix can also automatically refresh data and perform the masking process on a set schedule, ensuring that development and testing teams are always using masked copies of the latest data, with the referential integrity of the original data maintained.

Another benefit of Delphix is the ability to leverage intrinsic data replication and scheduling to make physical or virtual backups within data centers, between data centers, between data centers and clouds, and between clouds. Replication can be used to provide backup and granular restore capabilities down to the second, giving non-production data users a simple way to backup their environments. This task is often a very low priority for IT ops and DBA teams primarily responsible for protecting production data.

Delphix currently has no obvious competitors. After spending time onsite with the Delphix team and being able to provision and deploy data environments ourselves, we find that Delphix offers a unique opportunity to accelerate application projects, increase utilization of non-production data platforms, reduce capital expenses on storage hardware and software, all while reducing the direct costs of maintaining, replicating, and refreshing non-production data. Our hands-on test shows that Delphix delivers on its promises and that the software does work well. Any company with significant application development programs, IT operations efforts, and business users of production data should evaluate Delphix for solving problems that most IT professionals assume cannot be fixed.

About Storage Strategies NOW™

Storage Strategies NOW™ (SSG-NOW) is an industry analyst firm focused on storage, server, cloud and virtualization technologies. Our goal is to convey the business value of adopting these technologies to corporate stakeholders in a concise and easy-to-understand manner. Note: The information and recommendations made by Storage Strategies NOW are based upon public information and sources and may also include personal opinions both of Storage Strategies NOW and others, all of which we believe to be accurate and reliable. As market conditions change however, and not within our control, the information and recommendations are made without warranty of any kind. All product names used and mentioned herein are the trademarks of their respective owners. Storage Strategies NOW, Inc. assumes no responsibility or liability for any damages whatsoever (including incidental, consequential or otherwise), caused by your use of, or reliance upon, the information and recommendations presented herein, nor for any inadvertent errors which may appear in this document.