Chicago 2026

regoUniversity

Apptio • Targetprocess • Cloudability

Choosing the Right Integration Approach: Datalink Classic, Datalink, and ADM

Your Guides:

Chris Roades

Senior Apptio Consultant

Andrew Schneider

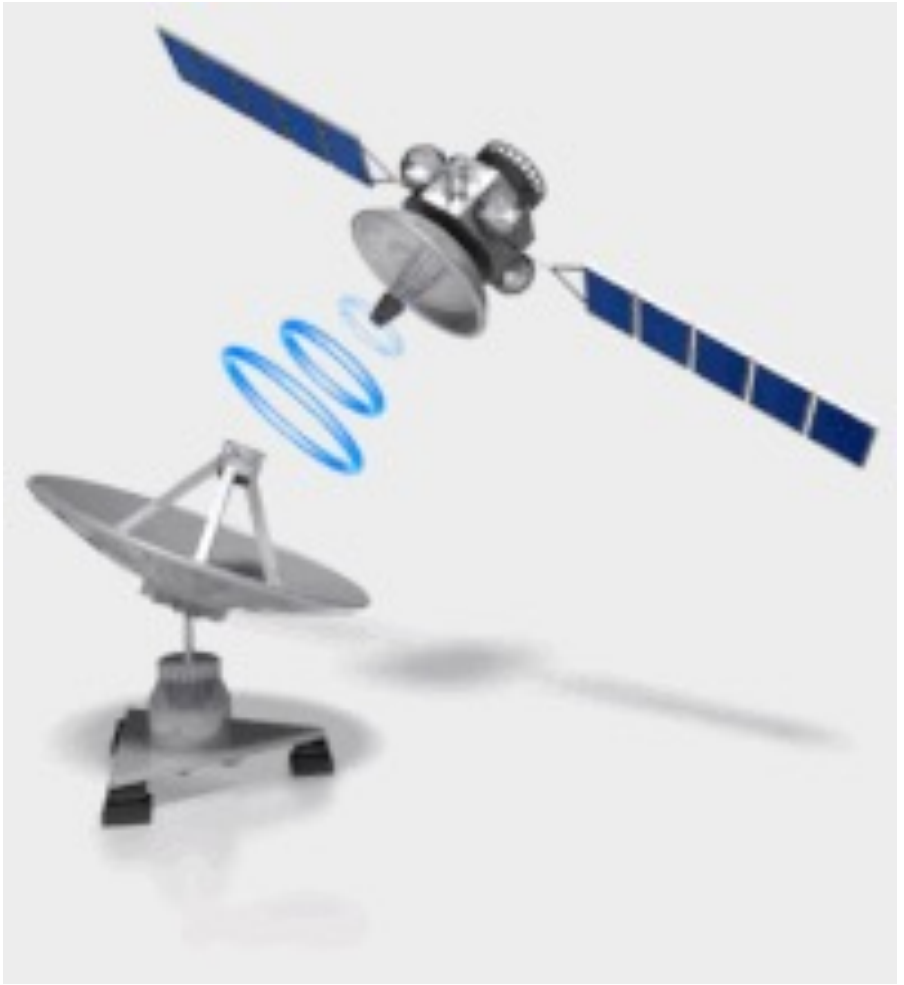
Senior Consultant - TBM

Agenda

- Introduction and Discussion Points
- Datalink Classic
  - Use cases, Limitations, Best Practices
- Datalink
  - Use cases, Limitations Best Practices
- Automated Data Management
  - Use cases, Limitations, Best Practices
- Choosing the right tool
- Technical Best Practices

Discussion Point

What are a few keys pieces of information needed to ensure a "seamless" data integration process?



Why Your Integration Approach Matters

Cost & Efficiency

Poor integration design creates recurring manual effort, delays data refreshes, and increases the total cost of ownership for your Apptio environment.

Maintainability

Integrations built without a clear framework become brittle over time — difficult to troubleshoot, document, and hand off to new team members.

Scalability

As data volumes and use cases grow (Costing, Planning, Targetprocess), the right foundation prevents costly re-architecture down the road.

Datalink Classic

Understanding the legacy file-based integration approach

Strengths & Ideal Use Cases

Strengths

- ✓ Simple to understand — no API knowledge required
- ✓ Works with virtually any data source that can export a file
- ✓ Low barrier to entry for teams new to Apptio
- ✓ Compatible with older on-premises Apptio deployments
- ✓ Useful for one-time or ad hoc data loads

Ideal Use Cases

One-Time Data Loads

Migrating historical data into Apptio during implementation or model changes.

Low-Frequency Updates

Data that changes quarterly or annually — not worth building a complex pipeline.

Legacy / On-Prem Environments

Where Datalink or ADM are not yet available or supported.

Proof of Concept

Quickly validating a data model before investing in a full integration build.

Limitations & When to Avoid It

No Native Automation

Scheduling requires external tools (cron, SFTP scripts, ETL). Any gap or failure in the external scheduler silently breaks the integration.

Schema Rigidity

Column order and naming must match exactly. Any upstream schema change in the source system breaks the integration until manually fixed.

No Built-In Monitoring

There's no native alerting when a file is late, malformed, or rejected. Teams rely on manual spot-checks to catch failures.

No Transformation Engine

Complex business logic must be applied before the file is created, pushing transformation work outside of Apptio entirely.

High Operational Overhead

File preparation, formatting, upload, and validation steps must be handled manually or with custom tooling — increasing ongoing effort.

Not Suitable for Targetprocess

ADM or native connectors are required for Targetprocess data; Datalink Classic has no integration path.

Best Practices (If You're Still Using It)

01

Lock Down the File Schema

Document the exact column names, order, data types, and null handling. Treat the schema as a contract between the source system and Apptio.

02

Build Validation Checks Before Upload

Add pre-upload scripts that check row counts, date ranges, and mandatory fields. Reject files that fail checks rather than letting Apptio silently skip rows.

03

Timestamp and Archive Every File

Retain dated copies of every file loaded. This provides an audit trail and allows re-loading historical data if the model changes.

04

Externalise Scheduling with Alerting

Don't rely on manual runs. Use a scheduler (e.g., Azure Data Factory, Control-M, cron) with failure notifications so missing uploads are caught immediately.

05

Document the End-to-End Process

Record the source, transformation logic, file location, schedule, and responsible owner. Integrations without documentation become orphaned and break silently.

06

Plan the Migration Path

If you're on Datalink Classic, start assessing Datalink or ADM. Build the migration into your roadmap before the manual overhead becomes critical.

Datalink

Apptio's structured integration framework

Strengths & Ideal Use Cases

Strengths

- ✓ Built-in scheduling — no external scheduler required
- ✓ File-based source connections in one tool
- ✓ Column mapping and basic transformation support
- ✓ Run history and load logs for audit and troubleshooting
- ✓ Reduces manual effort vs. Datalink Classic significantly

Ideal Use Cases

Recurring Costing Data Loads

Weekly or monthly cost data from GL, HR, and vendor systems into Apptio Cost Transparency.

Database Source Connections

Direct JDBC connections to source databases without intermediate file extraction steps.

Planning Data Feeds

Feeding budget and actuals data into Apptio IT Planning on a scheduled basis.

Moderate Complexity Pipelines

Integrations requiring column mapping, filtering, or simple transformations.

Limitations & When to Avoid It

Apptio Expertise Required

Configuration requires deep knowledge of Apptio data models and Datalink's connection syntax. Misconfigurations are hard to diagnose.

Error Visibility Gaps

Failures are logged, but there is no built-in alerting to notify teams when a scheduled run fails or produces unexpected results.

Limited API Support

While JDBC is supported, modern REST API integrations are limited. Connecting to SaaS sources often still requires an intermediary file step.

Limited Transformation Logic

Complex data reshaping — pivots, multi-source joins, derived calculations — must still be handled upstream before Datalink can consume the data.

No Cross-Product Coverage

Datalink does not natively support Targetprocess data loads — a separate integration approach is required for that product.

Maintenance as Models Evolve

Model changes in Apptio often require Datalink configurations to be manually updated to reflect new column names or structures.

Best Practices for Reliable Implementations

Name Connections Descriptively

Use a consistent naming convention: [Source]-[Target]-[Frequency]. Avoids confusion when managing multiple integrations in the same environment.

Limit Transformation Complexity

Keep transformation logic simple within Datalink. Push complex calculations upstream to the source or to a staging database where it can be tested independently.

Test in Non-Production First

Always validate new Datalink configurations in a development or UAT Apptio environment before pushing to production.

Version-Control Your Configurations

Export and store Datalink configuration files in source control. This enables rollback and provides documentation of what each integration does.

Set Up Email Alerts for Failures

Configure run failure notifications so the integration team is alerted immediately — don't rely on downstream data consumers to catch missed loads.

Automated Data Management (ADM)

Apptio's modern cloud-native integration platform

Strengths & Ideal Use Cases

Strengths

- ≈ Full cross-product coverage: Costing, Planning, Targetprocess
- ✓ Built-in scheduling, monitoring, and failure alerting
- ✓ Rich transformation engine — handles complex pipeline logic
- ✓ Significantly reduced operational overhead vs. older tools
- ✓ Cloud-native scalability — handles high data volumes reliably

Ideal Use Cases

Enterprise-Scale Integrations

High-volume, multi-source data pipelines feeding Costing, Planning, and Targetprocess simultaneously.

Targetprocess Data Loads

The only supported integration path for feeding work management data into Targetprocess.

New Apptio Implementations

All greenfield implementations should default to ADM as the integration standard.

Migrating from Legacy Tools

When reducing manual overhead and improving reliability are the primary drivers for modernisation.

Limitations & When to Avoid It

Higher Setup Investment

ADM requires more upfront configuration and design effort than simpler tools. Not the right choice for a one-time data load.

Steeper Learning Curve

Teams new to ADM need training before they can build and manage pipelines confidently. Factor in enablement time for new projects.

Cloud Deployment Required

ADM is a cloud-native tool. Organisations with strict on-premises requirements may face constraints on adoption.

Overkill for Simple Use Cases

If all you need is an occasional file load of flat data, ADM's full feature set adds unnecessary complexity.

Fewer Community Examples

As a newer platform, ADM has a smaller body of community documentation and examples compared to Datalink.

Migration Effort from Legacy

Moving existing Datalink or Classic integrations to ADM requires re-configuration and testing — it is not an automatic upgrade.

Best Practices for Scalable ADM Integrations

01

Design Pipelines Modularly

Break large integrations into smaller, reusable pipeline components. Modular design makes debugging faster and reuse easier across projects.

02

Leverage Built-In Validation

Use ADM's native schema validation and data quality checks at ingestion time. Catching bad data early prevents downstream model corruption.

03

Configure Alerting Before Go-Live

Set up failure and threshold alerts before the first production run. Never wait until a missed load is discovered by end users.

04

Use Staging Environments

Test all pipeline changes in a non-production environment first. ADM's power comes with risk — a misconfigured pipeline can load incorrect data at scale.

05

Document Each Pipeline

Record the source, schedule, transformation rules, responsible owner, and known failure modes for every ADM pipeline. Treat it as a living document.

06

Build a Migration Roadmap

If moving from Datalink or Classic, prioritise migrations by business impact. Run old and new pipelines in parallel during transition until validated.

Choosing the Right Tool

Decision framework, use case mapping, and common pitfalls

Decision Framework: Key Questions to Ask First

1	**Is this a one-time or recurring load?**	- One-time → Datalink Classic or manual upload may suffice - Recurring → Datalink or ADM
2	**Does it involve Targetprocess data?**	- Yes → ADM is required - No → any tool may apply
3	**What is the data volume and frequency?**	- High volume / daily+ → ADM preferred - Low volume / monthly → Datalink or Classic
4	**How complex is the transformation logic?**	- Simple mapping → Datalink - Complex / multi-source → ADM
5	**What is the operational overhead tolerance?**	- Low tolerance → ADM (built-in automation) - Manual OK → Datalink Classic

Use Case Mapping: Costing, Planning & Targetprocess

	DATALINK CLASSIC	DATALINK	ADM
COSTING	✓ One-time & ad hoc loads ✓ Simple flat data sources ✗ Not for automated pipelines	✓ Recurring cost data (GL, HR) ✓ DB-to-Apptio connections ~ Limited transformation	✓ Full automation support ✓ Complex multi-source loads ✓ Preferred for new builds
PLANNING	✓ One-off budget data imports ✗ No scheduling built in ✗ Manual effort every cycle	✓ Budget & actuals feeds ✓ Scheduled pipeline support ~ Basic transformation only	✓ End-to-end automation ✓ Complex planning pipelines ✓ Built-in monitoring
TARGETPROCESS	✗ Not supported	✗ Not supported	✓ Only supported path ✓ Work & resource data ✓ Real-time capable

Common Mistakes & How to Avoid Them

✗ Using Datalink Classic for recurring loads

→ Switch to Datalink or ADM. Classic requires manual effort every cycle — this creates failure risk and operational debt.

✗ No monitoring or alerting on any integration

→ Every integration must have failure notification. Silent failures are the most damaging kind.

✗ Choosing a tool by familiarity, not fit

→ Always run through the decision framework. Familiar tools chosen out of habit often become maintenance burdens.

✗ Skipping non-production testing

→ Always test in Dev/UAT before production. Even a small pipeline change can load incorrect data at scale.

✗ Building in Datalink when Targetprocess is in scope

→ Plan for ADM from day one. Retrofitting an integration approach mid-project is expensive.

✗ No documentation or ownership assigned

→ Every integration must have a named owner and documented spec before go-live. Undocumented integrations become orphaned.

Discussion Points

- To what extent is your Data Process managed via Datalink or Datalink Classic? Scheduled vs Manually Executed?

- Are there any major blockers you've encountered when leveraging or enabling more automation? What are they?

Technical Best Practices

Building reliable, scalable integrations across all three tools

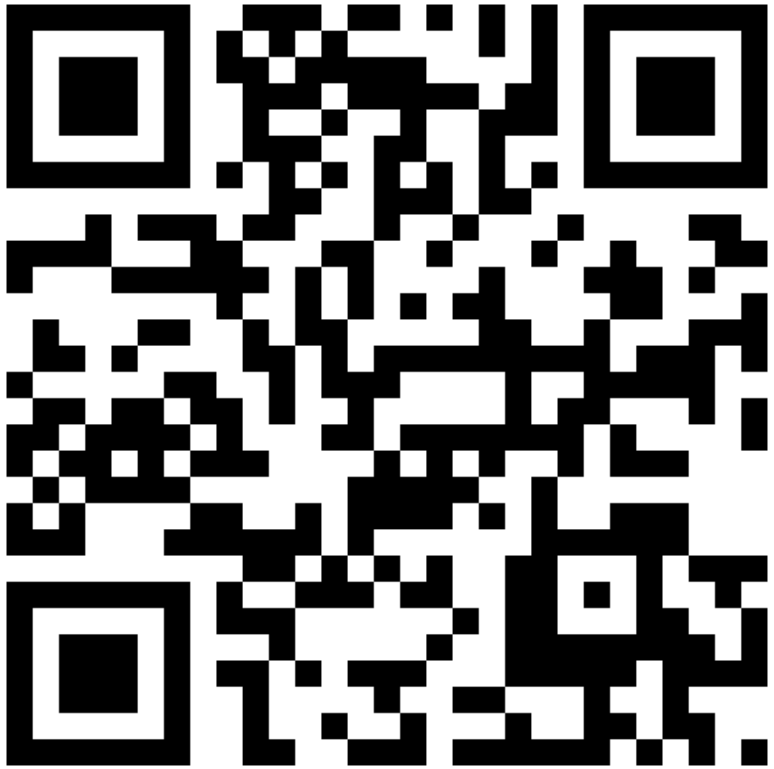
Best Practices

Data Quality & Validation	*Bad data in = bad decisions out. Validation must be enforced at every stage of the pipeline.*
Error Handling & Monitoring	*An integration without monitoring is a ticking clock. Build observability in — not as an afterthought.*
Scalability & Performance	*Design for the data volumes you'll have in two years — not the volumes you have today.*
Governance & Maintainability	*The integrations you build today will be maintained by someone else tomorrow. Design accordingly.*

Questions?

Surveys

Please take a few moments to fill out the class survey.
Your feedback is extremely important for future events.



Thank You For Attending Rego University

Instructions for PMI credits

- Access your account at pmi.org
- Click on **Certifications**
- Click on **Maintain My Certification**
- Click on **Visit CCR's** button under the **Report PDU's**
- Click on **Report PDU's**
- Click on **Course or Training**
- Class Provider = **Rego Consulting**
- Class Name = **regoUniversity**
- Course **Description**
- Date Started = **Today's Date**
- Date Completed = **Today's Date**
- Hours Completed = **1 PDU per hour of class time**
- Training classes = **Technical**
- Click on **I agree** and **Submit**

Let us know how we can improve!
Don't forget to fill out the class survey.

Phone
888.813.0444

Email
info@regoconsulting.com

Website
www.regouniversity.com