

Data 360 Testing Best Practices

Testing Data 360 effectively begins with deploying your configurations using data kits so each environment is aligned. Once deployed, SIs must load or simulate test data, since sandboxes provide metadata only. With configurations and data in place, the next step is to validate flows, rules, and behavior through a mix of automated and manual tests. Finally, you must review results, resolve any issues, and repeat the cycle across environments.

Deploying Data Kits

- **Use standard data kits to:**
 - Bundle and move Data 360 features from one environment to another.
- Deploy metadata and data processing logic across sandboxes using the local data kit deployment method.
- **Use DevOps data kits to:**
 - Migrate metadata from sandbox to production.
 - Make sure configurations, including governance rules, are consistent across environments.
 - Support controlled deployments by following the steps below:
 - Add components (e.g., data model objects, calculated insights) to a data kit.
 - Deploy the kit to the target environment using the local or DevOps deployment method.

How to Deploy Data 360 Between Environments

Deploying Data 360 between environments requires a mix of data kits, automation, and manual review. Some steps may be automated through CI/CD pipelines, while others require manual setup, review, and validation.

- **How deployments work:**
 - Data 360 uses data kits to package configurations such as metadata, rules, data flows, data models, and processing logic.
 - These kits are then deployed to another environment using either the local (manual) method or the DevOps (automated) method.
 - According to Salesforce, "For sandbox environments, local data kit deployment is a feature that allows replicating configurations within the same sandbox. This is limited to standard data kits and doesn't support DevOps data kits."
 - "Salesforce recommends using managed 2GP packages or manifestbased (a list of Data 360 metadata) deployments to transfer Data 360 metadata between environments. Unlocked 2GP packages are not supported for Data 360 components."
- **Automation vs. manual steps:**
 - Standard data kits are mostly manual and are used in early development or singlesandbox testing.
 - DevOps data kits allow CI/CD pipelines to move configurations automatically from Dev to QA, UAT, and then Prod.
 - Salesforce notes that "certain fixes, such as handling schema drift or adjusting environmentspecific endpoints, are still manual tasks even in automated pipelines."
- **What can be automated:**
 - Packaging and validating a data kit
 - Deploying kits to target environments
 - Triggering CI/CD pipelines
 - Running automated tests in Integration and QA
- **What must be done manually:**
 - Selecting the components to include in a kit
 - Approving deployments to higher environments
 - Loading or simulating test data (sandboxes contain metadata only)
 - Reviewing test results and validating behavior
 - Fixing schema drift, failed deployments, or hardcoded endpoints
 - Refreshing caches

Testing Data

Staging Testing Data Best Practices

- Sandbox metadata: When a sandbox is created, only metadata is copied over, so you must load or simulate your own data for testing.
- Use synthetic data for development and integration testing.
- Use anonymized production data for QA and UAT.
- Create small test datasets with known or expected results to verify data quality, data flow, and duplicate handling.
- Scale data volumes for performance validation.
- Use SQL queries or filters that retrieve only the necessary data.
- Add WHERE clauses to reduce unnecessary row processing.
- Select only the fields required to minimize memory and processing load.
- Ingest only the records needed for your Data 360 use cases.
- Pre-filter data before ingestion if full detail is unnecessary.
- Use a small but representative sample for functional testing.
- Only introduce full-scale datasets when specifically testing load or performance.
- Create standardized, reusable test datasets so results are comparable across environments and after sandbox refreshes.
- Test with incomplete data, unexpected formats, invalid values, and duplicates to expose hidden issues.
- Validate full sequences, not just data flow components.

Testing Frequency and Environment Best Practices

- Run tests in Developer and Developer Pro after every update or change.
- Run daily tests in Integration.
- Run QA and UAT each release cycle.
- Validate changes in a sandbox before deploying up the pipeline.
- Keep a shared record of endpoints, connection settings, data sources, and environment differences.
- Confirm that all required data model objects, rules, and data flows exist and are activated in the target environment.
- Use consistent naming conventions so testers know exactly what configuration was deployed.
- Use Data 360 logs and monitoring tools to identify failures, schema drift, and anomalies that functional testing may not reveal.
- Refreshes may remove test data or include legacy metadata. Perform a subsequent smoke test or full validation.
- Log issues by environment to identify patterns, such as drift, missing components, or connection problems that occur in specific orgs.

Common Pitfalls and Lessons Learned

- **Companion Connections:**
 - Avoid creating connections between production and the Data 360 home org before sandbox setup. Doing so can result in mismatched orgs and data.
- **Schema Drift:**
 - Monitor for schema drift, when the structure of the data changes in unexpected ways, which can break the data flow.
- **Fixed or Hard-Coded Endpoints:**
 - Avoid using fixed or hard-coded endpoints or URLs, as they only work in one environment and will break when moved to another environment.
- **Caches:**
 - Data flows and quality caches may require forced refreshes after changes.
- **Legacy Metadata:**
 - Remove legacy metadata from sandboxes created before August 2024. Sandboxes created before that date may contain old or outdated metadata.