

# Log Refactoring Phase 1 - Infrastructure Extraction

---

Status: IN\_PROGRESS

## Objective

Extract infrastructure and helper components from `Generic.cs` to improve modularity and reduce the monolithic footprint.

## Tasks

### 1. Communication Service Extraction (COMPLETED/REFACTORED)

- Identify redundant `callUrl` and `callUrlWithPayloadAsync` wrappers in `BaseObj.cs`.
- Instead of creating a new service, cleaned up `BaseObj.cs` by removing pass-through methods that merely delegated to `utils`.
- Updated `Generic.cs` to call `HttpService.CallUrl` directly, eliminating the unnecessary indirection layer.

### 2. Redis Service Extraction (COMPLETED)

- Encapsulate all `redisMan` calls into a `RedisService`.
- Define a clean interface for key/value and hash operations.
- **Refactored:** Moved semantic key construction from `RedisService` to `BaseObj` to eliminate redundant service layer.
- Eliminated `RedisService.cs` as it was absorbed by the `BaseObj` identity-aware implementation.

### 3. Data Serializer Extraction (IN\_PROGRESS)

- Analyzed current `DataSerializer.cs` (JSON) and `XmlDataSerializer.cs` (XML) in `IOB-WIN-FORM`.
- Decided to move these to `IOB_UT_NEXT` to make them available to the entire IOB hierarchy.
- Move files to `IOB_UT_NEXT\Iob\Services\`.
- Update namespaces and references.
- Integrate into `BaseObj.cs` via `protected` helper methods (e.g., `JsonSerialize<T>`, `XmlSerialize<T>`).
- Update `Generic.cs` to use the new `BaseObj` helpers.

### 4. BaseObj Simplification (COMPLETED)

- Analyze `BaseObj` responsibilities (State, Config, Messaging, Diagnostics).
- Remove pass-through methods for `utils` (Network, MAC, WebClients) to reduce "noise" in the base class.
- Cleaned up commented-out code blocks (`#if false`) to improve readability.

## Progress Log

- Created WIP document.

- Completed Redis Semantic Refactoring.
- Completed BaseObj Cleanup.
- Identified Data Serializer components for extraction.
- Started migration of Serializers to `IOB_UT_NEXT`.