



Elections Canada

Power Platform Development Standards

Application Lifecycle Management & Naming Conventions

Version 1.0

March 2025

Elections Canada Power Platform Development Standards

Table of Contents

- Elections Canada Power Platform Development Standards
 - Table of Contents
 - Introduction
 - Naming Conventions
 - Importance of Consistent Naming Standards
 - Standard Naming Patterns
 - Department Prefixes
 - Environment Types
 - Field Naming
 - Power Pages Site Naming
 - Environment Strategy
 - ALM Pipeline Implementation
 - Development Workflow
 - Azure DevOps Integration
 - Pipeline Stages
 - Power Pages Implementation
 - Standard Template
 - Authentication
 - Governance and Administration
 - Best Practices
 - Solution Management
 - Version Control
 - Testing
 - Documentation
 - Monitoring

Introduction

This document outlines the development standards and Application Lifecycle Management (ALM) practices for Elections Canada's Power Platform implementations, focusing on Model-Driven Apps and Power Pages portals. It combines Elections Canada's specific naming conventions with Microsoft's recommended healthy ALM practices to ensure consistent, maintainable, and secure applications.

Elections Canada has established a standardized approach for external-facing Power Pages portals, featuring accessibility tools, themed layouts, and Entra External ID SSO integration. This document serves as a comprehensive guide for developers working within the Elections Canada ecosystem to maintain consistency and follow best practices.

Naming Conventions

Importance of Consistent Naming Standards

Consistent naming standards are critical for the successful implementation and management of Power Platform resources at Elections Canada. These standards:

1. **Enhance Discoverability:** Makes it easier to locate resources across environments
2. **Improve Governance:** Supports effective resource management and access control
3. **Facilitate Collaboration:** Enables developers to understand and work with each other's components
4. **Streamline Support:** Simplifies troubleshooting and support processes
5. **Ensure Compliance:** Addresses regulatory and organizational requirements
6. **Reduce Development Time:** Eliminates confusion and rework due to inconsistent naming

Elections Canada has established the following naming standards for all Power Platform resources, as documented in the official [Cloudstrucc Public GitHub repository](#).

Standard Naming Patterns

Resource Type	Pattern	Example
Solutions	[Department]-[Project]-[Purpose]	EC-VoterReg-Core
Applications	[Department] [Purpose]	EC Voter Registration App
Entities/Tables	[Prefix]_[Entity]	ec_voter
Fields/Columns	[Prefix]_[Name]	ec_voterid
Forms	[Entity] - [Type]	Voter - Main
Views	[Entity]: [Purpose]	Voters: Active
Business Rules	BR: [Entity] - [Description]	BR: Voter - Validate Address
Workflows	WF: [Entity] - [Description]	WF: Voter - Send Confirmation
Flows	FL: [Entity/Area] - [Description]	FL: Voter - Process Registration
Canvas Apps	[Department]: [Purpose] [App]	EC: Voter Field App
Model-Driven Apps	[Department]: [Purpose]	EC: Voter Management
Environments	[Department]-[Project]-[Type]	EC-VoterReg-DEV
Connection References	CR: [Service] - [Purpose]	CR: SharePoint - Voter Documents
Cloud Flows	[Trigger Type]: [Description]	Auto: Process New Voter

Department Prefixes

- **EC:** Elections Canada
- **VO:** Voter Operations
- **FO:** Field Operations
- **PO:** Political Operations

- **EF:** Electoral Financing

Environment Types

- **DEV:** Development
- **TST:** Test
- **UAT:** User Acceptance Testing
- **PRD:** Production
- **SBX:** Sandbox

Field Naming

Field names should follow these additional rules:

- Use camelCase for field names
- Avoid spaces and special characters
- Use standard prefixes for specific field types:
 - **dt_** for Date fields
 - **num_** for Numeric fields
 - **opt_** for Option Set fields
 - **lkp_** for Lookup fields

Power Pages Site Naming

Power Pages sites should follow this convention:

```
[Department]-[Purpose]-[Environment].[domain].gc.ca
```

Example: **ec-voterportal-prd.elections.gc.ca**

Environment Strategy

Elections Canada implements a multi-environment strategy aligned with the Power Platform ALM best practices:

```

graph LR
    A[Developer Sandbox] --> B[Development]
    B --> C[Test/QA]
    C --> D[User Acceptance Testing]
    D --> E[Production]

    classDef dev fill:#d4f1f9,stroke:#333,stroke-width:1px
    classDef qa fill:#ffe6cc,stroke:#333,stroke-width:1px
    classDef prod fill:#d9ead3,stroke:#333,stroke-width:1px

    class A,B dev
    class C,D qa
    class E prod

```

Environment Purposes:

1. **Developer Sandbox:** Individual environments for developer experimentation
2. **Development:** Collaborative environment for ongoing development
3. **Test/QA:** Environment for testing by QA team
4. **UAT:** Environment for business stakeholder validation
5. **Production:** Live environment for end-users

ALM Pipeline Implementation

Elections Canada implements the Power Platform Pipelines feature with integration to Azure DevOps for automated build and deployment processes.

Development Workflow

```

flowchart TD
    A[Feature Branch Creation] --> B[Local Development]
    B --> C[Push Changes to Feature Branch]
    C --> D[Create Pull Request]
    D --> E[Code Review]
    E --> F[Automated Build Check]
    F --> G[Merge to Development Branch]
    G --> H[Deploy to DEV Environment]
    H --> I[QA Testing]
    I --> J[Deploy to TEST Environment]
    J --> K[UAT Testing]
    K --> L[Deploy to PROD Environment]

    classDef dev fill:#d4f1f9,stroke:#333,stroke-width:1px
    classDef review fill:#ffe6cc,stroke:#333,stroke-width:1px

```

```
classDef deploy fill:#d9ead3,stroke:#333,stroke-width:1px
classDef test fill:#fff2cc,stroke:#333,stroke-width:1px

class A,B,C dev
class D,E,F review
class G,H,J,L deploy
class I,K test
```

Azure DevOps Integration

Elections Canada leverages Azure DevOps for source control, work item tracking, and pipeline automation:

```
graph TD
    A[Azure DevOps Repository] --- B[Source Control]
    A --- C[Work Item Tracking]
    A --- D[Build Pipelines]
    A --- E[Release Pipelines]

    B --> F[Branch Policies]
    B --> G[Pull Request Validation]

    D --> H[Solution Packaging]
    D --> I[Automated Testing]

    E --> J[Environment Deployment]
    E --> K[Approval Gates]

    classDef repos fill:#d4f1f9,stroke:#333,stroke-width:1px
    classDef build fill:#ffe6cc,stroke:#333,stroke-width:1px
    classDef release fill:#d9ead3,stroke:#333,stroke-width:1px

    class A,B,C repos
    class D,H,I build
    class E,J,K release
```

Pipeline Stages

The ALM pipeline consists of several stages aligned with the environment strategy:

1. Build Stage

- Triggered by pull request or commit to main/development branch
- Packages solution files
- Validates solution against quality checks

- Generates build artifacts

2. Development Deployment

- Automatically deploys to development environment
- Runs automated tests
- Validates functionality

3. Test Deployment

- Requires manual approval from development lead
- Deploys to test environment
- Triggers notification to QA team

4. UAT Deployment

- Requires QA approval
- Deploys to UAT environment
- Triggers notification to business stakeholders

5. Production Deployment

- Requires business owner approval
- Scheduled during maintenance windows
- Includes backup procedures
- Post-deployment validation

```
sequenceDiagram
    participant Dev as Developer
    participant ADO as Azure DevOps
    participant PPP as Power Platform Pipelines
    participant Env as Environments

    Dev->>ADO: Commit code
    ADO->>PPP: Trigger build pipeline
    PPP->>PPP: Package solution
    PPP->>ADO: Store build artifacts
    ADO->>PPP: Trigger release pipeline
    PPP->>Env: Deploy to DEV
    PPP->>ADO: Notify QA for approval
    ADO->>PPP: QA approval granted
    PPP->>Env: Deploy to TEST
    PPP->>ADO: Notify stakeholders for UAT
    ADO->>PPP: UAT approval granted
    PPP->>Env: Deploy to PROD
    PPP->>ADO: Deployment complete notification
```


Power Pages Implementation

Elections Canada has established a standardized Power Pages portal template that is ready to deploy for any client within EC that requires an external-facing portal.

Standard Template

The standard Power Pages template includes:

- Accessibility tools integration
- EC-approved theme and branding
- Responsive design for all devices
- Content templates for common EC scenarios
- Standardized navigation structure

```
graph TD
    A[EC Power Pages Template] --> B[Accessibility Features]
    A --> C[EC Branding Theme]
    A --> D[Responsive Components]
    A --> E[Content Templates]
    A --> F[Navigation Structure]
    A --> G[Extra External ID SSO]

    B --> B1[Screen Reader Support]
    B --> B2[Keyboard Navigation]
    B --> B3[High Contrast Mode]

    C --> C1[EC Color Scheme]
    C --> C2[Typography Standards]
    C --> C3[Component Styling]

    G --> G1[User Registration]
    G --> G2[Authentication Flow]
    G --> G3[Profile Management]

    classDef core fill:#d4f1f9,stroke:#333,stroke-width:1px
    classDef features fill:#ffe6cc,stroke:#333,stroke-width:1px

    class A core
    class B,C,D,E,F,G features
```

Authentication

All external-facing Power Pages portals use Entra External ID for secure authentication:

```
sequenceDiagram
    participant User as External User
    participant PP as Power Pages Portal
    participant Entra as Entra External ID
    participant D365 as Dataverse

    User->>PP: Access portal
    PP->>User: Redirect to login
    User->>Entra: Authenticate
    Entra->>User: Authentication successful
    Entra->>PP: Return authentication token
    PP->>D365: Verify user permissions
    D365->>PP: Return user context
    PP->>User: Display personalized portal
```

Governance and Administration

Elections Canada implements a structured governance model for Power Platform development:

Center of Excellence (CoE)

- Establishes and enforces standards
- Reviews and approves new app requests
- Monitors environment usage and performance

Environment Administration

- Dedicated admin team for each environment
- Regular access reviews
- License management
- Data loss prevention policies

Security and Compliance

- Regular security reviews
- Compliance with EC security policies
- Data residency requirements
- Privacy impact assessments

Best Practices

Solution Management

- Use solution segmentation to separate core components from customizations
- Implement solution layering for extensibility
- Use managed solutions for all deployments beyond development
- Document dependencies between solutions

Version Control

- All customizations must be tracked in source control
- Feature branches for all new development
- Pull request reviews required before merging to main branches
- Tag all release versions in source control

Testing

- Unit tests for complex business rules
- UI automated tests for critical paths
- Load testing for high-volume processes
- User acceptance testing with documented test cases

Documentation

- Solution architecture documents
- Component-level documentation within source code
- User guides for end-users
- Administrative guides for support team

Monitoring

- Implement application insights
- Set up automated alerts for critical failures

- Regular performance reviews
 - Usage analytics to guide future development
-

This comprehensive approach to Power Platform development ensures Elections Canada can efficiently deliver high-quality, consistent, and secure solutions while maintaining governance and compliance requirements.