Contents

RADIANT Deployer Architecture & Deployment Packages **1**

Overview . . . 1
Deployment Modes . . . 1
Mode Definitions . . . 1
INSTALL Mode (Fresh Installation) . . . 2
UPDATE Mode (Upgrade Existing) . . . 2
ROLLBACK Mode (Revert to Previous) . . . 3
Deployment Package Structure . . . 3
Package Manifest . . . 4
Package Storage Locations . . . 4
Key Implementation Files . . . 5
Swift Deployer . . . 5
Build Tools . . . 5
Data Flow Diagrams . . . 6
Install Flow . . . 6
Update Flow . . . 6
Rollback Flow . . . 7
Verification Checklist . . . 7
Deployment Modes . . . 7
AI Registry Seeding . . . 7
Deployment Packages . . . 7
Parameter Rules . . . 7
Related Documentation . . . 8

RADIANT Deployer Architecture & Deployment Packages

Technical Architecture Document

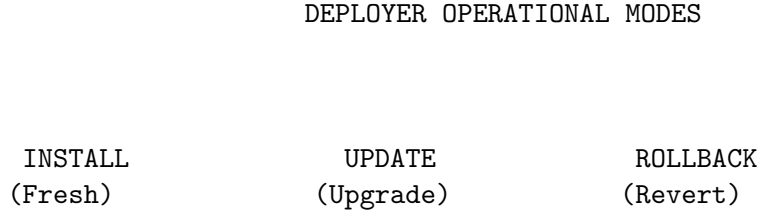
Version: 4.18.1 | Last Updated: December 2024

Overview

The RadiantDeployer Swift app operates in three distinct modes, each with different behaviors for parameter handling, package selection, and database operations.

Deployment Modes

Mode Definitions



```
Use Default          Read Current         Read Target
Parameters           From Instance        Snapshot


Seed AI              Merge User           Restore
Registry             Changes              Previous


Create New           Apply Delta          Apply
Instance             Changes              Snapshot
```

INSTALL Mode (Fresh Installation)

Trigger: No existing deployment detected for app/environment combination

Key Behaviors: 1. Uses DEFAULT parameters from tier configuration 2. Runs ALL database migrations (fresh) 3. SEEDS the AI Registry with providers and models 4. Creates initial admin user 5. Stores deployment metadata

Parameter Source: `InstallationParameters.defaults()`

```
// Parameters are initialized with tier-appropriate defaults
let parameters = InstallationParameters.defaults(
    appId: app.id,
    environment: environment,
    tier: .growth  // Based on selected tier
)
```

UPDATE Mode (Upgrade Existing)

Trigger: Existing deployment detected AND target version >= current version

Key Behaviors: 1. Fetches current parameters FROM the running instance 2. Creates pre-update snapshot for rollback 3. MERGES user changes with current parameters 4. Validates parameter changes are safe 5. Runs INCREMENTAL migrations only 6. **DOES NOT** seed AI Registry (preserves admin customizations)

Parameter Source: Running instance API + user modifications

```
// Parameters fetched from instance, then merged with user changes
```

```
let currentParameters = await fetchCurrentParameters(app, environment, credentials)
let updatedParameters = mergeParameters(current: currentParameters, changes: userChanges)
```

ROLLBACK Mode (Revert to Previous)

Trigger: User explicitly requests rollback OR target version < current version

Key Behaviors: 1. Loads target snapshot from S3 2. Creates safety snapshot of current state 3. Deploys with SNAPSHOT parameters (not current, not defaults) 4. Optionally restores database from RDS snapshot 5. Does not modify AI Registry

Parameter Source: Selected snapshot

Deployment Package Structure

Deployment packages are self-contained, versioned bundles containing everything needed to deploy a specific version of RADIANT.

```
radiant-4.18.0-abc123.radpkg
  manifest.json              # Package metadata & verification
  checksums.sha256           # File integrity verification

  infrastructure/            # CDK Stacks (compiled)
    cdk.out/                 # Synthesized CloudFormation
    lib/                     # CDK TypeScript (compiled)
    cdk.json                 # CDK configuration

  migrations/                # Database migrations
    radiant/                 # Core schema migrations
    thinktank/               # Think Tank specific
    seeds/                   # Seed data (AI Registry, etc.)

  functions/                 # Lambda function code
    api/                     # API handlers
    admin/                   # Admin handlers
    billing/                 # Billing handlers
    thermal/                 # Thermal management

  admin-dashboard/           # Next.js admin dashboard
    .next/                   # Compiled Next.js

  config/                    # Default configurations
    defaults.json            # Default parameters per tier
    providers.json           # AI provider seed data
    models.json              # AI model seed data
```

Package Manifest

```
{
  "packageFormat": "radpkg-v1",
  "version": "4.18.0",
  "buildId": "abc123def456",
  "buildTimestamp": "2024-12-24T10:30:00Z",

  "components": {
    "radiantPlatform": {
      "version": "4.18.0",
      "minUpgradeFrom": "4.15.0"
    },
    "thinkTank": {
      "version": "3.2.0",
      "minUpgradeFrom": "3.0.0"
    }
  },

  "compatibility": {
    "minimumDeployerVersion": "4.16.0",
    "supportedTiers": ["SEED", "STARTER", "GROWTH", "SCALE", "ENTERPRISE"],
    "supportedRegions": ["us-east-1", "us-west-2", "eu-west-1", "ap-southeast-1"]
  },

  "installBehavior": {
    "seedAIRegistry": true,
    "createInitialAdmin": true,
    "runFullMigrations": true
  },

  "updateBehavior": {
    "seedAIRegistry": false,
    "preserveAdminCustomizations": true,
    "runIncrementalMigrations": true,
    "createPreUpdateSnapshot": true
  }
}
```

Package Storage Locations

```
1. DEPLOYER APP CACHE (Local)
   ~/Library/Application Support/RadiantDeployer/packages/
     radiant-4.18.0-abc123.radpkg
     radiant-4.17.0-def456.radpkg
     index.json
```

```
2. S3 RELEASE BUCKET (Cloud - Official Releases)
   s3://radiant-releases-{region}/
      stable/
         radiant-4.18.0-abc123.radpkg
         latest.json
      beta/
         radiant-4.19.0-beta1-xyz789.radpkg
      archive/
          radiant-4.17.0-def456.radpkg

3. DEPLOYED INSTANCE (Cloud - Per Instance)
   s3://radiant-{appId}-{env}-deployments/
      current/
         radiant-4.18.0-abc123.radpkg
      snapshots/
          snapshot-2024-12-24T10-30-00Z/
             package.radpkg
             parameters.json
             db-snapshot-id.txt
          ...
```

Key Implementation Files

Swift Deployer

File	Purpose
`Models/InstallationParameters.swift`	DeploymentMode enum, TierLevel, InstallationParameters, InstanceParameters, ParameterChanges, DeploymentSnapshot
`Services/DeploymentService.swift`	Mode detection, executeInstall, executeUpdate, executeRollback, parameter fetching/merging
`Services/PackageService.swift`	Package discovery, download, verification, caching
`Views/Deployment/ParameterEditorView.swift`	UI for editing parameters based on mode

Build Tools

File	Purpose
`tools/scripts/build-package.sh`	Build deployment packages from source
`tools/version-manager.ts`	Version bumping and synchronization

Data Flow Diagrams

Install Flow

```
User                    Deployer                    AWS


1. Select App
2. Select Env
3. Select Tier
4. Click Deploy
                        5. Check instance exists      (None found)
                        6. Mode = INSTALL
                        7. Load DEFAULT params
                        8. Download latest package      S3
                        9. Verify package integrity
                        10. Deploy CDK stacks       CloudFormation
                        11. Run ALL migrations       Aurora
                        12. SEED AI Registry        Aurora
                        13. Create initial admin      Cognito
                        14. Store deployment meta     S3 + DB
                        15. Report success
```

Update Flow

```
User                    Deployer                    AWS + Instance


1. Select App
2. Select Env
3. Change Params
4. Click Update
                        5. Check instance exists      (Found!)
                        6. Mode = UPDATE
                        7. Fetch CURRENT params       Radiant API
                        8. Create snapshot         S3
                        9. MERGE user changes
                        10. Validate changes
                        11. Download target package     S3
                        12. Deploy CDK stacks       CloudFormation
                        13. Run INCREMENTAL migrations   Aurora
                        14. SKIP AI Registry seeding
                        15. Update deployment meta     S3 + DB
                        16. Report success
```

Rollback Flow

```
User                    Deployer                    AWS


1. Select App
2. Select Env
3. Select Snapshot
4. Click Rollback
                        5. Mode = ROLLBACK
                        6. Load target snapshot       S3
                        7. Validate compatibility
                        8. Create safety snapshot     S3
                        9. Download snapshot package    S3
                        10. Deploy CDK stacks       CloudFormation
                        11. Optionally restore DB     RDS Snapshot
                        12. Update deployment meta     S3 + DB
                        13. Report success
```

Verification Checklist

Deployment Modes

- **On INSTALL**: Parameters come from defaults
- **On UPDATE**: Parameters come from running instance + user changes
- **On ROLLBACK**: Parameters come from selected snapshot

AI Registry Seeding

- AI Registry is seeded ONLY on fresh install
- On UPDATE, AI Registry is preserved (not touched)
- Admins can add/remove providers via Admin Dashboard

Deployment Packages

- Packages are created by build-package.sh script
- Package creation is triggered by code changes or version bumps
- Packages are stored in local cache, S3 release bucket, and instance bucket

Parameter Rules

- Region CANNOT be changed after install
- Tier CAN be changed on update (with feature validation)
- All parameter changes are tracked via snapshots

Related Documentation

- Deployer Admin Guide - User-facing documentation
- Deployment Guide - Deployment procedures
- API Reference - API documentation