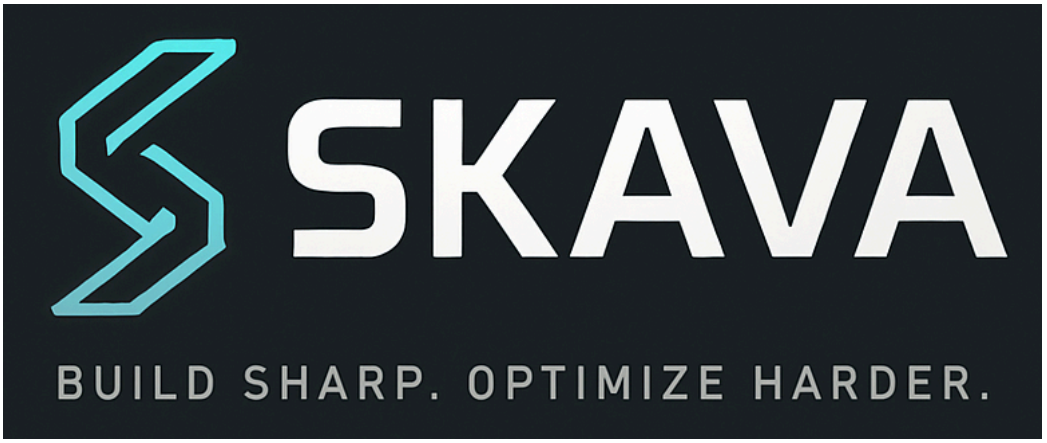

Asset Optimizer Pro - Documentation

Table of Contents

1. Getting Started
2. Installation
3. User Interface
4. Workflow
5. Asset Types
6. Optimization Profiles
7. Advanced Features
8. Best Practices
9. Troubleshooting
10. API Reference

Getting Started

System Requirements

- Unity 2019.4 or higher
- Windows, macOS, or Linux
- 4GB RAM minimum (8GB recommended)

Quick Start Guide

1. Import Asset Optimizer Pro from the Unity Asset Store
2. Open the tool via `Window > Asset Optimizer Pro`
3. Select your target platform
4. Click "Start Scanning"
5. Review and select assets to optimize

6. Click "Optimize" to apply changes

Installation

Step 1: Import the Package

1. Open Unity Package Manager
2. Search for "Asset Optimizer Pro"
3. Click "Import"
4. Select all files and click "Import"

Step 2: Initial Setup

Window → Asset Optimizer Pro

Step 3: Configure Settings

Navigate to the Settings tab to configure:

- Backup preferences
- Auto-scan options
- Default optimization profiles

User Interface

Main Window Components

1. Header Section

- Product branding and version
- Quick stats display
- Navigation controls

2. Workflow Steps

Visual progress indicator showing:

- **Setup**: Configure scan parameters
- **Scan**: Analyze project assets
- **Select**: Choose assets to optimize
- **Optimize**: Apply optimizations
- **Complete**: View results

3. Content Area

Dynamic content based on current workflow step

4. Footer

- Settings link
- Help documentation
- Version information

Workflow

Step 1: Setup

Configure your optimization parameters:

- **Target Platform**: Mobile, VR, Desktop, Console, or WebGL
- **Asset Types**: Select which types to scan
- **Quick Presets**: One-click configuration for common scenarios

Step 2: Scanning

The scanner analyzes your project:

- Identifies optimization opportunities
- Calculates potential savings
- Groups assets by type
- Provides detailed metrics

Step 3: Selection

Review and select assets:

- Filter by type or name
- Sort by size or savings potential
- Bulk selection tools
- Preview optimization impact

Step 4: Optimization

Apply optimizations:

- Preview changes before applying
- Real-time progress tracking
- Automatic backup creation
- Detailed logging

Step 5: Results

View optimization results:

- Total space saved
- Performance improvements
- Export detailed reports

- Start new scan

Asset Types

Textures

Supported Formats: PNG, JPG, JPEG, TGA, BMP, PSD, TIFF, GIF, EXR

Optimization Options:

- Maximum texture size
- Compression quality
- Format conversion
- Mipmap generation
- Platform-specific formats

Best Practices:

- Use power-of-two dimensions
- Enable compression for most textures
- Disable Read/Write when not needed
- Use appropriate formats per platform

Models

Supported Formats: FBX, OBJ, DAE, 3DS, BLEND

Optimization Options:

- Mesh compression levels
- Vertex optimization
- Animation compression
- Material import settings
- LOD generation hints

Best Practices:

- Remove unnecessary materials
- Optimize mesh topology
- Use appropriate compression
- Disable unused features

Audio

Supported Formats: WAV, MP3, OGG, AIFF

Optimization Options:

- Compression format

- Quality settings
- Sample rate optimization
- Mono conversion
- Load type configuration

Best Practices:

- Use compressed formats
- Convert to mono when possible
- Optimize sample rates
- Choose appropriate load types

Animations

Optimization Options:

- Keyframe reduction
- Compression settings
- Precision adjustments
- Curve optimization

Best Practices:

- Remove redundant keyframes
- Use appropriate precision
- Compress when possible

Optimization Profiles

Default Profiles

Mobile Profile

- Aggressive texture compression
- Reduced texture sizes (1024px max)
- High mesh compression
- Mono audio conversion
- Optimized for app store limits

VR Profile

- Balanced quality/performance
- Medium texture sizes (2048px)
- Stereo audio preservation
- Higher animation quality
- GPU instancing enabled

Desktop Profile

- High quality settings
- Large texture support (4096px)
- Minimal compression
- Full audio quality
- Advanced shader features

Console Profile

- Platform-specific optimizations
- Streaming texture support
- Balanced compression
- High-quality audio
- Optimized loading

WebGL Profile

- Maximum compression
- Small texture sizes
- Streaming audio
- Reduced polygon counts
- Web-optimized formats

Custom Profiles

Create custom profiles via the Profile Editor:

1. Click "Profiles" tab
2. Click "Create New Profile"
3. Configure settings
4. Save with descriptive name

Advanced Features

Batch Processing

Process multiple folders or asset types:

csharp
```
// Example: Batch optimize all textures
AssetOptimizerPro.BatchOptimize(AssetType.Texture, profile);
```

Command Line Interface

Automate optimization in build pipelines:

bash
```
Unity -batchmode -executeMethod AssetOptimizerPro.CLI.Optimize -platform mobile
```

Custom Rules

Create optimization rules using the API:

```
csharp
var rule = new OptimizationRule();
rule.AddCondition(asset => asset.size > 1024 * 1024);
rule.AddAction(asset => asset.compress = true);

optimizer.AddRule(rule);
```

Integration with Build Pipeline

```
csharp
public class OptimizeBuildProcessor : IPreprocessBuildWithReport
{
    public void OnPreprocessBuild(BuildReport report)
    {
        AssetOptimizerPro.OptimizeForPlatform(report.platform);
    }
}
```

Best Practices

1. Regular Optimization

- Run optimization before each build
- Set up automated scans
- Monitor asset growth

2. Platform-Specific Settings

- Use appropriate profiles
- Test on target devices
- Consider platform limitations

3. Quality vs Size Balance

- Preview all changes
- Test visual quality
- Keep backups of originals

4. Team Workflow

- Share optimization profiles
- Document exceptions
- Use version control

Troubleshooting

Common Issues

Q: Textures look blurry after optimization A: Adjust compression quality in texture settings or use a higher max resolution.

Q: Build size didn't decrease much A: Check if assets are actually included in build. Some may be referenced by Resources or Addressables.

Q: Optimization is taking too long A: Reduce batch size in settings or optimize by asset type.

Q: Can't find optimized assets A: Check backup folder. Enable "Show optimized assets" in settings.

Error Messages

"Failed to optimize asset"

- Check file permissions
- Ensure asset is not corrupted
- Try reimporting the asset

"Out of memory"

- Reduce batch size
- Close other applications
- Increase Unity memory allocation

API Reference

Core Classes

AssetOptimizer

Main optimization engine

csharp
```
public class AssetOptimizer
{
    public float Progress { get; }
    public string CurrentAsset { get; }

    public void OptimizeAssets(
        List<ScannedAsset> assets,
        OptimizationProfile profile,
        OptimizationProgressCallback callback
```

```
    );
}
```

AssetScannerSystem

Project scanning functionality

```
csharp
public class AssetScannerSystem
{
    public void ScanProject(
        Dictionary<AssetType, bool> typeFilters,
        ScanProgressCallback callback
    );
}
```

OptimizationProfile

Configuration settings

```
csharp
public class OptimizationProfile
{
    public string name;
    public PlatformTarget targetPlatform;
    public TextureOptimizationSettings textureSettings;
    public ModelOptimizationSettings modelSettings;
    public AudioOptimizationSettings audioSettings;
}
```

Extension Methods

```
csharp
// Check if asset needs optimization
bool needsOpt = asset.RequiresOptimization();

// Get optimization suggestions
var suggestions = asset.GetOptimizationSuggestions();

// Apply profile to selection
selectedAssets.ApplyProfile(mobileProfile);
```

Events

```
csharp
AssetOptimizerPro.OnScanComplete += (assets) => {
    Debug.Log($"Found {assets.Count} assets");
```

```
};

AssetOptimizerPro.OnOptimizationComplete += (report) => {
    Debug.Log($"Saved {report.totalSavings} bytes");
};
```