Contents

RADIANT Intelligent File Conversion Service

Version: 4.18.55
Last Updated: December 2024
Status: Production Ready

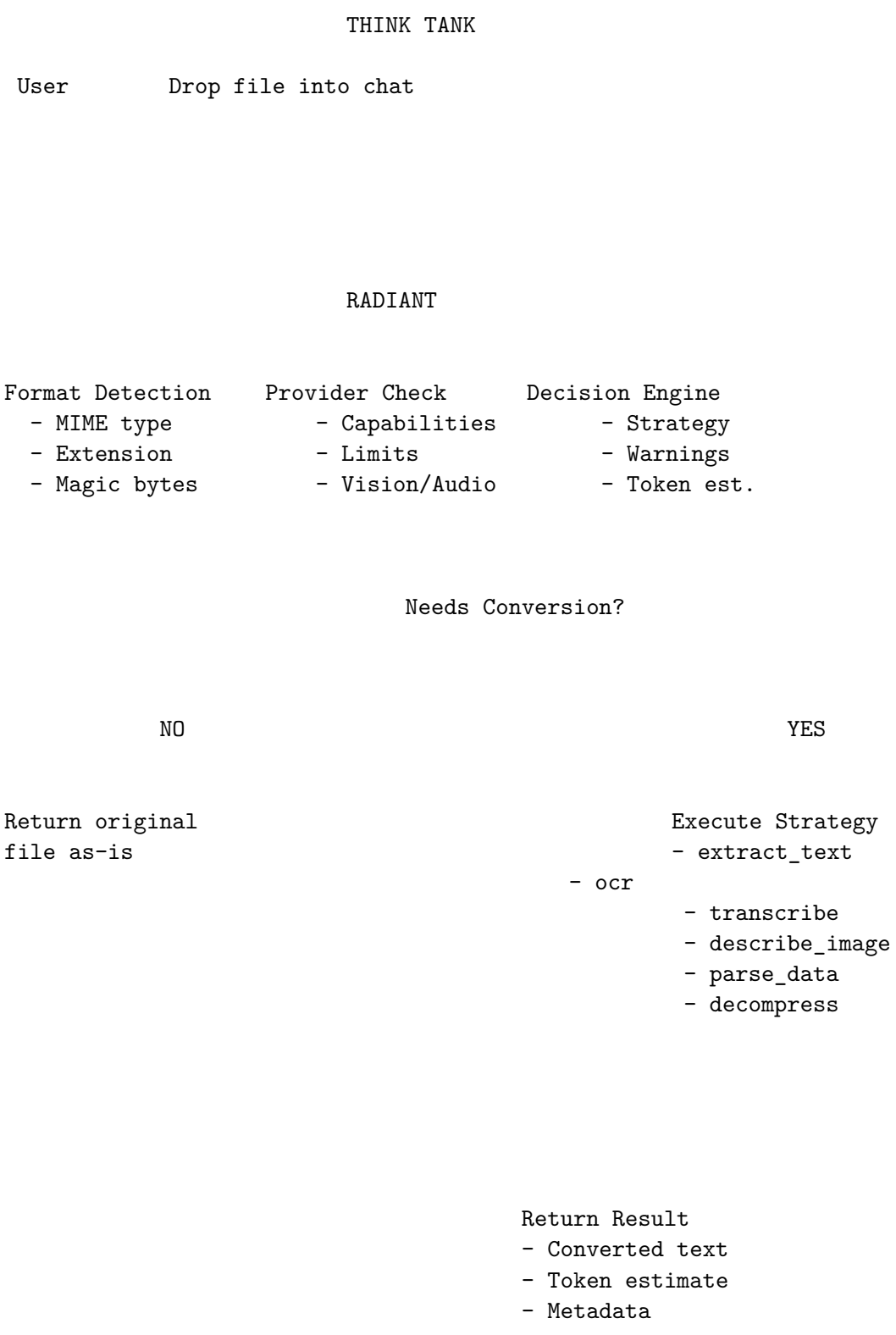
Overview

The **Intelligent File Conversion Service** is a Radiant-side system that automatically decides when and how to convert files for AI providers. The core principle is **"Let Radiant decide, not Think Tank"** - Think Tank simply drops files, and Radiant determines the optimal conversion strategy based on the target AI provider's capabilities.

Key Principles

1. **Think Tank submits files without worrying about provider compatibility**
2. **Radiant detects file format and checks target provider capabilities**
3. **Conversion only happens if the AI provider doesn't understand the format**
4. **Uses AI + libraries for intelligent conversion**

Architecture



Supported File Formats

Documents

Format	Extension	MIME Type	Conversion Strategy
PDF	`.pdf`	`application/pdf`	`extract_text` via pdf-parse
Word	`.docx`, `.doc`	`application/vnd.openxmlformats-officedocument.wordprocessingml`	`extract_text` via mammoth
PowerPoint	`.pptx`, `.ppt`	`application/vnd.openxmlformats-officedocument.presentationml.p`	`extract_text` via officeparser
Excel	`.xlsx`, `.xls`	`application/vnd.openxmlformats-officedocument.spreadsheetml.sh`	`parse_data` via officeparser

Text Files

Format	Extension	MIME Type	Notes
Plain Text	`.txt`	`text/plain`	Direct passthrough
Markdown	`.md`	`text/markdown`	Direct passthrough
JSON	`.json`	`application/json`	Direct or `parse_data`
CSV	`.csv`	`text/csv`	`parse_data`
XML	`.xml`	`application/xml`	Direct or `extract_text`
HTML	`.html`	`text/html`	`extract_text`

Images

Format	Extension	MIME Type	Conversion Strategy
PNG	`.png`	`image/png`	Native or `describe_image`
JPEG	`.jpg`, `.jpeg`	`image/jpeg`	Native or `describe_image`
GIF	`.gif`	`image/gif`	Native or `describe_image`
WebP	`.webp`	`image/webp`	Native or `describe_image`
SVG	`.svg`	`image/svg+xml`	Convert to PNG or `describe_image`
BMP	`.bmp`	`image/bmp`	Convert to PNG or `describe_image`
TIFF	`.tiff`	`image/tiff`	Convert to PNG or `describe_image`

Audio

Format	Extension	MIME Type	Conversion Strategy
MP3	`.mp3`	`audio/mpeg`	`transcribe` via Whisper
WAV	`.wav`	`audio/wav`	`transcribe` via Whisper
OGG	`.ogg`	`audio/ogg`	`transcribe` via Whisper
FLAC	`.flac`	`audio/flac`	`transcribe` via Whisper
M4A	`.m4a`	`audio/mp4`	`transcribe` via Whisper

Video

Format	Extension	MIME Type	Conversion Strategy
MP4	`.mp4`	`video/mp4`	`describe_video` - frame extraction
WebM	`.webm`	`video/webm`	`describe_video` - frame extraction
MOV	`.mov`	`video/quicktime`	`describe_video` - frame extraction
AVI	`.avi`	`video/x-msvideo`	`describe_video` - frame extraction

Code Files

Format	Extension	Notes
Python	`.py`	Syntax-highlighted markdown
JavaScript	`.js`, `.jsx`	Syntax-highlighted markdown
TypeScript	`.ts`, `.tsx`	Syntax-highlighted markdown
Java	`.java`	Syntax-highlighted markdown
C/C++	`.c`, `.cpp`, `.h`	Syntax-highlighted markdown
Go	`.go`	Syntax-highlighted markdown
Rust	`.rs`	Syntax-highlighted markdown
Ruby	`.rb`	Syntax-highlighted markdown

Archives

Format	Extension	MIME Type	Conversion Strategy
ZIP	`.zip`	`application/zip`	`decompress` - extract contents
TAR	`.tar`	`application/x-tar`	`decompress` - extract contents
GZIP	`.gz`, `.tar.gz`, `.tgz`	`application/gzip`	`decompress` - extract contents

Provider Capabilities

The service maintains a registry of AI provider capabilities:

Provider	Vision	Audio	Video	Max File Size	Native Document Formats
OpenAI	GPT-4V	Whisper		20MB	txt, md, json, csv
Anthropic	Claude 3			32MB	pdf, txt, md, json, csv
Google	Gemini			100MB	pdf, txt, md, json, csv
xAI	Grok			20MB	txt, md, json
DeepSeek				10MB	txt, md, json, csv
Self-hosted	LLaVA	Whisper		50MB	txt, md, json, csv

Conversion Strategies

1. `none` - No Conversion

Provider natively supports the format. File is passed through as-is.

2. `extract_text` - Text Extraction

Extracts plain text from documents using: - **PDF**: `pdf-parse` library - extracts all text, page metadata - **DOCX/DOC**: `mammoth` library - preserves structure, extracts images - **PPTX/PPT**: Text extraction from slides - **HTML/XML**: Strip tags, preserve content

Example output:

```
[Document Title]
Page 1:
Content from first page...

Page 2:
Content from second page...

[Metadata]
Pages: 10
Author: John Doe
Created: 2024-01-15
```

3. `ocr` - Optical Character Recognition

Uses AWS Textract to extract text from images containing text.

Features: - Detects printed and handwritten text - Table detection and extraction - Form field detection - Confidence scores per block

Example output:

```
[OCR Result]
Confidence: 94.5%

INVOICE #12345
Date: January 15, 2024

Item          Qty    Price
Widget A       10    $50.00
Widget B        5    $25.00

Total: $625.00
```

4. `transcribe` - Audio Transcription

Uses OpenAI Whisper API or self-hosted Whisper for speech-to-text.

Features: - Automatic language detection - Timestamp segments - SRT/VTT subtitle generation - Speaker diarization (future)

Example output:

```
[Transcription]
Duration: 5:32
Language: English
Model: whisper-1

[00:00] Hello and welcome to today's meeting.
[00:05] We'll be discussing the Q4 roadmap.
[00:12] First, let's review the current status...
```

5. `describe_image` - AI Image Description

Uses vision-capable models to describe image contents.

Supported Models: - GPT-4 Vision (OpenAI) - Claude 3 Vision (Anthropic) - LLaVA (self-hosted)

Features: - Detailed scene description - Text detection (OCR integration) - Object identification - Color and composition analysis

Example output:

```
[Image Description]
Model: gpt-4-vision
Dimensions: 1920x1080

This image shows a modern office space with an open floor plan.
In the foreground, there are several desks arranged in clusters,
each with monitors and office supplies. The walls are painted in
a neutral gray tone with large windows providing natural light.

[Text detected in image]:
"RADIANT - Innovation Center"
"Welcome Visitors"
```

6. `describe_video` - Video Frame Analysis

Extracts key frames from video and describes each using vision models.

Features: - Configurable frame interval (default: 10 seconds) - Maximum frames limit (default: 10) - Frame-by-frame descriptions - Narrative summary generation

Example output:

```
**Video Overview** (2m 30s, 1920x1080)

**Frame Analysis:**

**[0:00]** The video opens with a title screen showing the company logo
against a blue gradient background.
```

```
**[0:10]** A presenter in business attire stands in front of a whiteboard
with diagrams showing the system architecture.

**[0:20]** Close-up of the whiteboard showing a flowchart with boxes
labeled "User Input", "Processing", and "Output".

...

**Summary:**
The video begins with: Company logo and title screen
The video ends with: Presenter summarizing key points with bullet list
```

7. `parse_data` - Structured Data Parsing

Converts spreadsheets and data files to JSON.

Supported formats: - CSV → JSON array of objects - XLSX/XLS → JSON with sheet data - JSON → Validated and prettified

Example output (CSV):

```json
{
  "data": [
    {"name": "Alice", "email": "alice@example.com", "role": "Admin"},
    {"name": "Bob", "email": "bob@example.com", "role": "User"},
    {"name": "Carol", "email": "carol@example.com", "role": "User"}
  ],
  "metadata": {
    "rowCount": 3,
    "columnCount": 3,
    "headers": ["name", "email", "role"]
  }
}
```

Example output (Excel):

```json
{
  "sheets": [
    {
      "name": "Sales Data",
      "rows": [...],
      "headers": ["Date", "Product", "Revenue"],
      "rowCount": 150
    },
    {
      "name": "Summary",
      "rows": [...],
      "headers": ["Metric", "Value"],
      "rowCount": 10
    }
```

```
  ],
  "metadata": {
    "sheetCount": 2,
    "totalRows": 160,
    "hasFormulas": true
  }
}
```

8. `decompress` - Archive Extraction

Extracts and processes archive contents.

Supported formats: - ZIP (via adm-zip) - TAR (via tar) - GZIP (via zlib)

Features: - Recursive extraction - Text file content inclusion - Binary file detection - Size limits enforcement

Example output:

```
**Archive Contents** (ZIP)

**File Structure:**

  project/  project/README.md (2.5KB)  project/package.json (1.2KB)  project/src/
project/src/index.ts (5.3KB)  project/src/utils.ts (3.1KB)


**File Contents:**

### project/README.md

```markdown
# My Project

This is a sample project...
```

project/package.json

```
{
  "name": "my-project",
  "version": "1.0.0"
}
```


9. `render_code` - Code Formatting
Formats code files with syntax highlighting.

Example output:
````
```markdown
```typescript
import { Injectable } from '@angular/core';
```
```

```
@Injectable()
export class DataService {
  private data: string[] = [];

  getData(): string[] {
    return this.data;
  }
}
```
“‘

---

## API Reference

### Base Path

`/api/thinktank/files`

### Endpoints

### Process File

`POST /api/thinktank/files/process`

**Request:**

```
{
  "filename": "document.pdf",
  "mimeType": "application/pdf",
  "content": "<base64-encoded-content>",
  "targetProvider": "anthropic",
  "targetModel": "claude-3-5-sonnet",
  "conversationId": "conv-uuid-optional"
}
```

**Response:**

```
{
  "success": true,
  "data": {
    "conversionId": "conv_abc123",
    "originalFile": {
      "filename": "document.pdf",
      "format": "pdf",
      "size": 1048576,
      "checksum": "sha256:abc123..."
    },
    "convertedContent": {
      "type": "text",
      "content": "Extracted document text...",
      "tokenEstimate": 2500,
```

```
      "metadata": {
        "originalFormat": "pdf",
        "conversionStrategy": "extract_text",
        "pageCount": 10,
        "title": "Annual Report 2024",
        "author": "Finance Team"
      }
    },
    "processingTimeMs": 1250
  }
}
```


### Check Compatibility

`POST /api/thinktank/files/check-compatibility`

**Request:**


```
{
  "filename": "image.png",
  "mimeType": "image/png",
  "fileSize": 524288,
  "targetProvider": "deepseek"
}
```


**Response:**


```
{
  "success": true,
  "data": {
    "fileInfo": {
      "filename": "image.png",
      "format": "png",
      "size": 524288
    },
    "provider": {
      "id": "deepseek",
      "supportsFormat": false,
      "supportsVision": false,
      "maxFileSize": 10485760
    },
    "decision": {
      "needsConversion": true,
      "strategy": "describe_image",
      "reason": "Provider deepseek lacks vision - will use AI to describe image",
      "targetFormat": "txt",
      "warnings": []
    }
  }
}
```

### Get Provider Capabilities

```
GET /api/thinktank/files/capabilities
GET /api/thinktank/files/capabilities?provider=anthropic
```

**Response:**


```
{
  "success": true,
  "data": [
    {
      "providerId": "anthropic",
      "supportedFormats": ["png", "jpg", "jpeg", "gif", "webp", "pdf", "txt", "md", "json", "cs
      "nativeDocumentFormats": ["pdf", "txt", "md", "json", "csv"],
      "maxFileSize": 33554432,
      "supportsVision": true,
      "supportsAudio": false,
      "supportsVideo": false,
      "supportsDocuments": true
    }
  ]
}
```


### Get Conversion History

```
GET /api/thinktank/files/history
GET /api/thinktank/files/history?conversationId=conv-uuid&limit=50&offset=0
```

**Response:**


```
{
  "success": true,
  "data": {
    "conversions": [
      {
        "id": "conv_abc123",
        "filename": "report.pdf",
        "originalFormat": "pdf",
        "originalSize": 1048576,
        "targetProvider": "anthropic",
        "needsConversion": true,
        "strategy": "extract_text",
        "status": "completed",
        "tokenEstimate": 2500,
        "processingTimeMs": 1250,
        "createdAt": "2024-12-31T00:00:00Z"
      }
    ],
    "pagination": {
      "limit": 50,
```

```
      "offset": 0
    }
  }
}
```

### Get Conversion Statistics

```
GET /api/thinktank/files/stats
GET /api/thinktank/files/stats?days=30
```

**Response:**

```
{
  "success": true,
  "data": {
    "totalFiles": 1250,
    "convertedCount": 890,
    "nativeCount": 360,
    "failedCount": 12,
    "totalBytesProcessed": 2147483648,
    "avgProcessingMs": 850,
    "mostCommonFormat": "pdf",
    "mostCommonStrategy": "extract_text",
    "periodDays": 30
  }
}
```

---

## Database Schema

### Tables

**`file_conversions`** Tracks all file conversion decisions and results.

| Column | Type | Description |
|---|---|---|
| `id` | UUID | Primary key |
| `tenant_id` | UUID | Tenant reference |
| `filename` | VARCHAR(500) | Original filename |
| `original_format` | VARCHAR(50) | Detected format |
| `original_size` | BIGINT | File size in bytes |
| `target_provider` | VARCHAR(100) | Target AI provider |
| `target_model` | VARCHAR(200) | Target model ID |
| `needs_conversion` | BOOLEAN | Whether conversion was needed |
| `strategy` | VARCHAR(50) | Conversion strategy used |
| `conversion_status` | VARCHAR(20) | pending, processing, completed, failed |
| `converted_token_estimate` | INTEGER | Estimated tokens |
| `processing_time_ms` | INTEGER | Processing duration |
| `created_at` | TIMESTAMPTZ | Creation timestamp |

**provider_file_capabilities** Registry of provider file format support.

| Column | Type | Description |
|---|---|---|
| provider_id | VARCHAR(100) | Provider identifier (unique) |
| supported_formats | JSONB | Array of supported formats |
| native_document_formats | JSONB | Formats provider handles natively |
| max_file_size | BIGINT | Maximum file size in bytes |
| supports_vision | BOOLEAN | Has vision capabilities |
| supports_audio | BOOLEAN | Has audio capabilities |
| supports_video | BOOLEAN | Has video capabilities |

---

## Configuration

### Environment Variables

| Variable | Description | Default |
|---|---|---|
| FILE_CONVERSION_BUCKET | S3 bucket for file storage | radiant-files |
| OPENAI_API_KEY | OpenAI API key for Whisper/Vision | Required |
| ANTHROPIC_API_KEY | Anthropic API key for Claude Vision | Optional |
| WHISPER_ENDPOINT_URL | Self-hosted Whisper endpoint | Optional |
| VISION_ENDPOINT_URL | Self-hosted vision endpoint | Optional |

### Admin Configuration

**Location**: Admin Dashboard → Think Tank → File Settings

| Setting | Default | Description |
|---|---|---|
| Max file size | 50MB | Maximum upload size |
| Conversion timeout | 30s | Processing timeout |
| Enable transcription | true | Audio → text |
| Enable OCR | true | Image text extraction |
| Enable video processing | false | Video frame extraction |
| Retention days | 30 | How long to keep converted files |

---

## Implementation Files

| File | Purpose |
|---|---|
| lambda/shared/services/file-conversion-service.ts | Main service with decision engine |
| lambda/shared/services/converters/pdf-converter.ts | PDF text extraction |
| lambda/shared/services/converters/docx-converter.ts | DOCX/DOC text extraction |
| lambda/shared/services/converters/excel-converter.ts | Excel/CSV parsing |

| File | Purpose |
|---|---|
| `lambda/shared/services/converters/audio-converter.ts` | Audio transcription |
| `lambda/shared/services/converters/image-converter.ts` | Image conversion & OCR |
| `lambda/shared/services/converters/video-converter.ts` | Video frame extraction |
| `lambda/shared/services/converters/archive-converter.ts` | Archive extraction |
| `lambda/shared/services/converters/index.ts` | Module exports |
| `lambda/thinktank/file-conversion.ts` | API handlers |
| `migrations/127_file_conversion_service.sql` | Database schema |

---

## Dependencies

### NPM Packages

```
{
  "pdf-parse": "^1.1.1",
  "mammoth": "^1.6.0",
  "xlsx": "^0.18.5",
  "sharp": "^0.33.2",
  "fluent-ffmpeg": "^2.1.2",
  "adm-zip": "^0.5.10",
  "tar": "^6.2.0"
}
```

### AWS Services

- **S3**: File storage
- **Textract**: OCR processing
- **Lambda**: Processing execution

---

## Error Handling

### Common Errors

| Error | Cause | Resolution |
|---|---|---|
| `File size exceeds limit` | File > provider max | Reduce file size or extract portions |
| `Unsupported format` | Unknown file type | Convert to supported format first |
| `OCR failed` | Textract error | Check image quality, retry |
| `Transcription failed` | Whisper error | Check audio quality, verify API key |
| `PDF is password protected` | Encrypted PDF | Provide unencrypted version |

### Error Response Format

```
{
  "success": false,
  "error": "PDF extraction failed: File is password protected",
  "conversionId": "conv_abc123",
  "originalFile": {
    "filename": "protected.pdf",
    "format": "pdf",
    "size": 1048576
  },
  "processingTimeMs": 150
}
```

---

## Security Considerations

1. **File Size Limits**: Enforced per provider to prevent resource exhaustion
2. **Format Validation**: Magic bytes + extension verification
3. **Tenant Isolation**: RLS policies on all tables
4. **S3 Encryption**: AES-256 at rest
5. **Signed URLs**: Time-limited access to stored files
6. **Input Sanitization**: All filenames and metadata sanitized

---

## Monitoring

### Metrics

- Total files processed per tenant
- Conversion success/failure rate
- Average processing time
- Most common formats
- Most common conversion strategies
- Storage usage

### Alerts

- High failure rate (>5%)
- Processing time > 30s
- Storage quota approaching limit

---

---

## Domain-Specific File Formats

The service includes a comprehensive registry of domain-specific file formats that are widely used in specialized fields but not commonly supported by mainstream AI providers.

**Mechanical Engineering / CAD**

| Format | Extensions | Description | Library |
|---|---|---|---|
| **STEP** | `.step`, `.stp`, `.p21` | ISO 10303 CAD exchange | OpenCASCADE, FreeCAD |
| **STL** | `.stl` | 3D printing mesh | numpy-stl, trimesh |
| **OBJ** | `.obj` | Wavefront 3D model | trimesh, three.js |
| **Fusion 360** | `.f3d`, `.f3z` | Autodesk parametric CAD | Fusion 360 API |
| **IGES** | `.iges`, `.igs` | Legacy CAD exchange | OpenCASCADE |
| **DXF** | `.dxf` | AutoCAD 2D drawings | ezdxf |
| **GLTF/GLB** | `.gltf`, `.glb` | Web 3D format | three.js, trimesh |

**Electrical Engineering**

| Format | Extensions | Description | Library |
|---|---|---|---|
| **KiCad** | `.kicad_pcb`, `.kicad_sch` | PCB/schematic | kicad-cli, kiutils |
| **EAGLE** | `.brd`, `.sch` | Autodesk PCB | eagle-to-kicad |
| **SPICE** | `.spice`, `.sp`, `.cir` | Circuit simulation | PySpice, ngspice |

**Medical/Healthcare**

| Format | Extensions | Description | Library |
|---|---|---|---|
| **DICOM** | `.dcm`, `.dicom` | Medical imaging | pydicom, dcmtk |
| **HL7 FHIR** | `.json`, `.xml` | Health records | fhir.resources |

**Scientific/Research**

| Format | Extensions | Description | Library |
|---|---|---|---|
| **NetCDF** | `.nc`, `.nc4` | Climate/geoscience | netCDF4, xarray |
| **HDF5** | `.h5`, `.hdf5` | Scientific data | h5py |
| **FITS** | `.fits` | Astronomy data | astropy |

**Geospatial**

| Format | Extensions | Description | Library |
|---|---|---|---|
| **Shapefile** | `.shp`, `.dbf` | Vector GIS | geopandas, shapefile |
| **GeoTIFF** | `.tif`, `.geotiff` | Georeferenced raster | rasterio |

**Bioinformatics**

| Format | Extensions | Description | Library |
|---|---|---|---|
| **FASTA** | `.fasta`, `.fa` | DNA/protein sequences | Biopython |
| **PDB** | `.pdb` | Protein structure | Biopython, py3Dmol |

---

### Multi-Model File Preparation

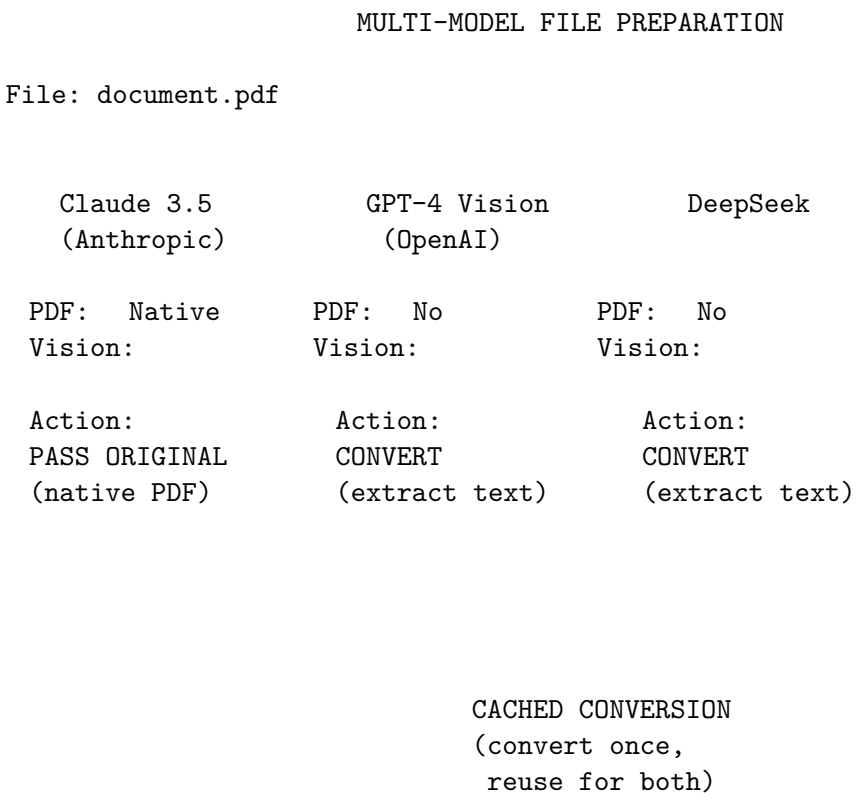
When multiple AI models work on the same prompt (multi-model orchestration), the system makes **per-model conversion decisions**:

#### Key Principle

> **"If a model accepts the file type, assume it understands it unless proven otherwise."**

- Only convert for models that don't support the format
- Pass original file to models with native support
- Cache conversions to avoid redundant processing

#### How It Works



### Per-Model Actions

| Action | When | Result |
|---|---|---|
| `pass_original` | Model natively supports format | Original file passed |
| `convert` | Model doesn't support format | Converted content passed |
| `skip` | File too large or conversion failed | Model excluded |

### Usage Example

```
import { multiModelFilePrepService } from './multi-model-file-prep.service';

// Prepare file for 3 models
const result = await multiModelFilePrepService.prepareFileForModels({
  tenantId,
  userId,
  file: {
    content: pdfBuffer,
    filename: 'document.pdf',
    mimeType: 'application/pdf',
  },
  targetModels: [
    { modelId: 'claude-3-5-sonnet', providerId: 'anthropic' },
    { modelId: 'gpt-4-vision', providerId: 'openai' },
    { modelId: 'deepseek-chat', providerId: 'deepseek' },
  ],
});

// Result:
// - Claude: pass_original (native PDF support)
// - GPT-4: convert (no PDF support, extract text)
// - DeepSeek: convert (reuses cached conversion)

// Get content for each model
for (const model of result.perModelPrep) {
  if (model.action !== 'skip') {
    const content = multiModelFilePrepService.getContentForModel(result, model.modelId);
    // Use content.data with this model
  }
}
```

### Model Format Overrides

When a model claims to support a format but proves it doesn't understand it well, overrides can be added:

```
// If Claude struggles with complex PDFs despite claiming support
multiModelFilePrepService.addFormatOverride(
  'claude-3-haiku',
  'pdf',
  'Struggles with multi-column PDFs'
);
// Now Claude 3 Haiku will get converted PDFs instead of originals
```

---

## AGI Brain Integration

The AGI Brain automatically detects domain-specific files and selects appropriate conversion strategies.

### How It Works

1. **File Detection**: When a file is uploaded, the system checks if it's a domain-specific format
2. **Domain Context**: The user's domain (from profile or conversation) influences strategy selection
3. **Library Selection**: The AGI Brain selects the best library based on availability and capabilities
4. **Conversion Planning**: A conversion plan is created with fallback strategies
5. **Execution**: The conversion is executed using the selected library

### Conversion Strategy Selection

The AGI Brain considers: - **User's domain**: Technical users get more detailed extraction - **Conversation context**: "show me a preview" → visual output, "export data" → structured data - **File complexity**: Simple formats get direct parsing, complex ones may need external tools - **Available libraries**: Falls back if preferred library isn't available

### Example: CAD File Processing

```
// AGI Brain detects a STEP file
const plan = planDomainConversion(
  'assembly.step',
  'application/step',
  'mechanical_engineering',  // User's domain
  'Can you analyze this CAD model?'  // Conversation context
);

// Returns:
{
  format: { format: 'step', domain: 'mechanical_engineering', ... },
  selectedStrategy: { strategy: 'extract_geometry', outputFormat: 'text', ... },
  selectedLibrary: { name: 'OpenCASCADE', pythonPackage: 'OCC', ... },
  requiresExternalService: true,
  estimatedComplexity: 'complex'
}
```

### AI Description Prompts

Each domain format includes a specialized AI prompt for when the AGI needs to describe the file without full parsing:

```
// STL file prompt
"This is an STL 3D model file. Describe the shape, identify what object
it might be, assess printability, and note any potential issues for 3D printing."

// DICOM file prompt
"This is a DICOM medical image. Describe the imaging modality, anatomical
region, and any visible findings. Note: Do not provide medical diagnoses."

// STEP file prompt
"This is a STEP CAD file. Describe the mechanical part or assembly,
including approximate geometry, features (holes, fillets, chamfers),
and likely manufacturing process."
```

---

## Implementation Files

| File | Purpose |
|---|---|
| `lambda/shared/services/file-conversion-service.ts` | Main service with decision engine |
| `lambda/shared/services/converters/pdf-converter.ts` | PDF text extraction |
| `lambda/shared/services/converters/docx-converter.ts` | DOCX/DOC text extraction |
| `lambda/shared/services/converters/excel-converter.ts` | Excel/CSV parsing |
| `lambda/shared/services/converters/audio-converter.ts` | Audio transcription |
| `lambda/shared/services/converters/image-converter.ts` | Image description & OCR |
| `lambda/shared/services/converters/video-converter.ts` | Video frame extraction |
| `lambda/shared/services/converters/archive-converter.ts` | Archive extraction |
| `lambda/shared/services/converters/cad-converter.ts` | CAD/3D file parsing (STL, OBJ, STEP, DXF, GLTF) |
| `lambda/shared/services/converters/domain-formats.ts` | Domain format registry (50+ formats) |
| `lambda/shared/services/converters/agi-brain-integration.ts` | AGI Brain integration |
| `lambda/shared/services/converters/index.ts` | Module exports |
| `lambda/thinktank/file-conversion.ts` | API handlers |
| `migrations/127_file_conversion_service.sql` | Database schema |

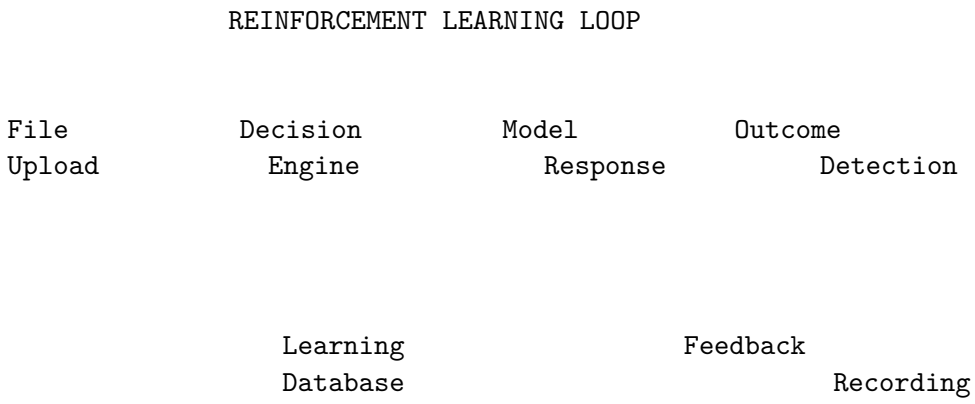
---

---

## Reinforcement Learning Integration

The file conversion system integrates with the AGI Brain/consciousness for persistent learning from conversion outcomes.

### How Learning Works




### What Gets Learned

| Signal | Source | Learning |
|---|---|---|
| **User Rating** | Explicit feedback (1-5 stars) | Direct quality signal |
| **Model Response** | Auto-inferred from response text | Did model understand? |
| **Error Detection** | Model errors/hallucinations | Format incompatibility |
| **Conversion Success** | Pass original worked | Model handles format |
| **Conversion Failure** | Pass original failed | Model needs conversion |

### Understanding Score

Each model/format combination has an understanding score (0.0 to 1.0):

| Score | Meaning | Action |
|---|---|---|
| 0.8 - 1.0 | Excellent understanding | Pass original |
| 0.6 - 0.8 | Good understanding | Pass original |
| 0.4 - 0.6 | Moderate understanding | May convert |
| 0.0 - 0.4 | Poor understanding | Convert |

### Learning Database Schema

**Migration:** `128_file_conversion_learning.sql`

| Table | Purpose |
|---|---|
| `model_format_understanding` | Per-tenant model/format understanding scores |
| `conversion_outcome_feedback` | Recorded feedback for learning |
| `format_understanding_events` | Audit trail of score changes |
| `global_format_learning` | Cross-tenant aggregate insights |

### Recording Feedback

```
import { fileConversionLearningService } from './file-conversion-learning.service';

// Record outcome after model responds
await fileConversionLearningService.recordOutcomeFeedback({
  tenantId,
  userId,
  conversionId: 'conv_abc123',
  modelId: 'claude-3-5-sonnet',
  providerId: 'anthropic',
  filename: 'document.pdf',
  fileFormat: 'pdf',
  actionTaken: 'pass_original',
  outcome: 'success',  // or 'partial', 'failure'
  outcomeSource: 'user_feedback',
  userRating: 5,
  modelUnderstood: true,
});

// Result: Understanding score updated, learning candidate created if significant
```

### Auto-Inference from Response

The system can automatically infer outcomes from model responses:

```
const inference = fileConversionLearningService.inferOutcomeFromResponse(
  modelResponse,
  'pdf'
);

// Returns:
// {
//   outcome: 'failure',
//   modelUnderstood: false,
//   modelMentionedFormatIssues: true,
//   confidence: 0.8
// }
```

**Failure signals detected:** - “I can’t read”, “unable to process”, “cannot access the file” - “appears to be empty”, “binary data”, “base64” - Model asking for clarification about file content

### Integration with Consciousness

Significant learning events create **Learning Candidates** for the consciousness system:

| Event | Learning Candidate Type | Quality |
|---|---|---|
| Model failed on format it claimed to support | `format_misunderstanding` | 0.85 |
| Unnecessary conversion (model would have understood) | `unnecessary_conversion` | 0.70 |
| Model hallucinated file content | `hallucination_detection` | 0.90 |
| User gave negative rating | `user_correction` | 0.85 |

These feed into the LoRA evolution system for persistent consciousness improvement.

### Admin Override

Admins can force conversion regardless of learning:

```
// Force conversion for a model/format that consistently fails
await fileConversionLearningService.setForceConvert(
  tenantId,
  'claude-3-haiku',
  'pdf',
  'Struggles with multi-column PDFs',
  adminUserId
);

// Clear override
await fileConversionLearningService.clearForceConvert(
  tenantId,
  'claude-3-haiku',
  'pdf'
);
```

### Implementation Files

| File | Purpose |
|---|---|
| `lambda/shared/services/file-conversion-learning.service.ts` | Learning service |
| `migrations/128_file_conversion_learning.sql` | Database schema |

---

### Related Documentation

- THINKTANK-ADMIN-GUIDE.md - Section 27
- RADIANT-ADMIN-GUIDE.md