Aras Innovator Integration Project Guide

Document Purpose

This document provides complete project context for Claude Code to continue development of the Aras Innovator integration project. It includes project overview, completed work, technical specifications, and next phases.

Table of Contents

---

1. Project Overview

1.1 Purpose

Implement a complete PLM/PDM system integration for a client using:

- **PLM System**: Aras Innovator Community Edition
- **CAD System**: CATIA V5-6R2022
- **ERP System**: SAP ERP

1.2 Project Phases

Phase	Description	Status
Phase 1	Windows Explorer "Send To" (Check-in tool)	✅ **COMPLETED**
Phase 2	Standalone Windows App (Search, Browse, Check-out)	⏭️ SKIPPED
Phase 3	**CATIA Add-in (Full Integration)**	🔄 **NEXT**
Phase 4	SAP ERP Integration	⏳ PENDING

1.3 Architecture Overview

```
┌─────────────────┐   ┌─────────────────┐   ┌─────────────────┐
│          │   │          │   │          │
│  CATIA V5-6   │◄───►│ Integration   │◄───►│   Aras    │
│  R2022 SP6   │ COM │  Layer    │ IOM │  Innovator   │
│          │ API │  (C# .NET)   │ API │          │
└─────────────────┘   └────────┬────────┘   └─────────────────┘
                 │
                 │ RFC/OData
                 ▼
          ┌─────────────────┐
          │          │
          │  SAP ERP    │
          │          │
          └─────────────────┘
```

---

2. Client Environment

2.1 Software Versions

Software	Version	Notes
Aras Innovator	Community Edition 14+	Installed & configured
CATIA	P3 V5-6R2022 SP6	Platform 3 (full features)
SAP ERP	TBD	To be confirmed

Software	Version	Notes
Windows	Windows 10/11	Client workstations
.NET Framework	4.7.2+	Development target
Visual Studio	2022	Development IDE

2.2 CATIA Details

Aspect	Value
Product	CATIA P3 V5-6R2022 SP6
Platform Level	P3 (Full features)
Equivalent Version	V5 R32
API Type	COM Automation
Supported Languages	C#, VB.NET, VBA
File Types	.CATPart, .CATProduct, .CATDrawing

2.3 Aras Innovator Details

Aspect	Value
Version	Community Edition 14+
Server URL	http://localhost/InnovatorServer (default)
Database	InnovatorSolutions (default)
API	IOM (Innovator Object Model)
Authentication	Username/Password

2.4 Development Environment

Tool	Purpose
Visual Studio 2022	IDE for C# development

Tool	Purpose
Git	Version control
GitHub	Repository hosting
Claude Code	AI-assisted development
.NET Framework 4.7.2	Target framework

---

3. Completed Work

3.1 Phase 1: Windows Explorer Send To Tool

Repository: (Your GitHub repository URL)

Features Implemented:

- ✅ Right-click file → Send To → Aras Innovator
- ✅ Login dialog with credential saving
- ✅ Document creation in Aras
- ✅ File upload to Aras vault
- ✅ Multi-file selection support
- ✅ Configuration saved to AppData

Project Structure:

```
ArasSendTo/
├── ArasSendTo.sln
├── ArasSendTo.csproj
├── Program.cs
├── ConfigManager.cs
├── ArasService.cs
├── Forms/
│   ├── LoginForm.cs
│   ├── LoginForm.Designer.cs
│   ├── MainForm.cs
│   └── MainForm.Designer.cs
├── Properties/
└── Lib/
    └── IOM.dll
```

Key Classes:

Class	Purpose
Program.cs	Entry point, receives file paths as arguments
ConfigManager.cs	Load/save settings as JSON to AppData
ArasService.cs	Connect, login, create document, upload file
LoginForm.cs	Server URL, database, username, password
MainForm.cs	File list, document properties, upload progress

3.2 Aras PDM Configuration

Completed:

- ✅ Aras Innovator installed
- ✅ Admin login verified
- ✅ Default data model explored
- ✅ Test parts and documents created
- ✅ BOM structure tested

PDM Setup Guide: See `Aras_PDM_Setup_Guide.md`

4. Phase 3: CATIA Add-in Development

4.1 Overview

Build a CATIA V5 add-in that integrates directly with Aras Innovator, allowing users to manage PLM data without leaving CATIA.

4.2 Features Required

Feature	Priority	Description
Login	High	Connect to Aras from CATIA
Check-in	High	Save current CATPart/CATProduct to Aras
Check-out	High	Open file from Aras for editing (locked)
Get Latest	High	Download latest version (read-only)
Search	Medium	Find parts/documents in Aras
BOM Sync	Medium	Extract CATProduct assembly → Aras BOM
Properties Sync	Medium	Map CATIA properties ↔ Aras attributes
Lifecycle	Low	View/change item state
Where Used	Low	Show parent assemblies

4.3 Architecture

```
┌─────────────────────────────────────────────────────────────────┐
│ CATIA V5-6R2022                                  │
├─────────────────────────────────────────────────────────────────┤
│ [File] [Edit] [View] [Insert] [Tools] [Aras ▼] [Window] [Help] │
│                          ├────────────┤       │
│                          │ 🔐 Login    │      │
│                          │ 📥 Check-out │      │
│                          │ 📤 Check-in  │      │
│                          │ 🔄 Get Latest│      │
│                          │ 🔍 Search    │      │
│                          │ 📋 BOM Sync  │      │
│                    │ ⚙️ Settings │     │
│                    └─────────────┘        │
└──────────────────────────────────────────────────────────────────────┘
```

4.4 Technical Approach

Option A: CATIA Add-in (COM-based)

Pros:

- Full integration with CATIA toolbar
- Professional appearance
- Best user experience

Cons:

- More complex development
- Requires CATIA type library registration

Technology:

- C# Class Library
- COM Interop
- CATIA V5 Automation API
- Aras IOM API

Option B: External App with CATIA Automation

Pros:

- Easier to develop
- No CATIA installation needed for building

Cons:

- Separate window from CATIA
- Less integrated feel

Recommendation: Start with Option B for faster development, then migrate to Option A for production.

4.5 Project Structure

```
ArasCatiaIntegration/
├── ArasCatiaIntegration.sln
├── ArasCatiaAddin/                  # Main add-in project
│   ├── ArasCatiaAddin.csproj
│   ├── CatiaApplication.cs          # CATIA connection wrapper
│   ├── ArasService.cs               # Aras connection (reuse from Phase 1)
│   ├── ConfigManager.cs             # Settings (reuse from Phase 1)
│   ├── Commands/
│   │   ├── LoginCommand.cs
│   │   ├── CheckInCommand.cs
│   │   ├── CheckOutCommand.cs
│   │   ├── GetLatestCommand.cs
│   │   ├── SearchCommand.cs
│   │   └── BomSyncCommand.cs
│   ├── Forms/
│   │   ├── LoginForm.cs
│   │   ├── SearchForm.cs
│   │   ├── CheckInForm.cs
│   │   ├── BomSyncForm.cs
│   │   └── SettingsForm.cs
│   ├── Models/
│   │   ├── CatiaDocument.cs
│   │   ├── ArasDocument.cs
│   │   ├── BomItem.cs
│   │   └── PropertyMapping.cs
│   └── Utilities/
│       ├── FileHelper.cs
│       ├── BomExtractor.cs
│       └── PropertyMapper.cs
├── Lib/
│   ├── IOM.dll                      # Aras IOM
│   └── Interop.CATIA.dll            # CATIA interop (generated)
└── Documentation/
    └── UserGuide.md
```

4.6 CATIA COM API Reference

Connecting to CATIA

```
csharp
```

```
using INFITF;  // CATIA Infrastructure
using MECMOD;  // Mechanical Modeler
using PARTITF; // Part Interfaces
using ProductStructureTypeLib; // Assembly

// Connect to running CATIA instance
Type catiaType = Type.GetTypeFromProgID("CATIA.Application");
Application catia = (Application)Activator.CreateInstance(catiaType);

// Or get existing instance
Application catia = (Application)Marshal.GetActiveObject("CATIA.Application");
```

Getting Active Document

```csharp
// Get active document
Document activeDoc = catia.ActiveDocument;

// Check document type
if (activeDoc is PartDocument partDoc)
{
    Part part = partDoc.Part;
    // Work with part...
}
else if (activeDoc is ProductDocument prodDoc)
{
    Product product = prodDoc.Product;
    // Work with assembly...
}
```

Extracting BOM from CATProduct

```csharp
```

```
public List<BomItem> ExtractBom(Product rootProduct)
{
    var bomItems = new List<BomItem>();
    Products children = rootProduct.Products;

    for (int i = 1; i <= children.Count; i++)
    {
        Product child = children.Item(i);
        bomItems.Add(new BomItem
        {
            PartNumber = child.PartNumber,
            Name = child.Name,
            Quantity = 1, // Need to count occurrences
            FilePath = child.ReferenceProduct.Parent.FullName
        });
    }

    return bomItems;
}
```

Getting Document Path

```csharp
string filePath = activeDoc.FullName;  // Full path including filename
string fileName = activeDoc.Name;      // Just filename
```

4.7 Aras IOM API Reference

Connecting to Aras

```csharp
```

```
using Aras.IOM;

// Create connection
HttpServerConnection connection = IomFactory.CreateHttpServerConnection(
    serverUrl,      // "http://localhost/InnovatorServer"
    database,       // "InnovatorSolutions"
    username,       // "admin"
    password        // "innovator"
);

// Login
Item loginResult = connection.Login();
if (loginResult.isError())
{
    throw new Exception(loginResult.getErrorString());
}

// Get Innovator instance
Innovator innovator = loginResult.getInnovator();
```

Creating a Document

```csharp
Item doc = innovator.newItem("Document", "add");
doc.setProperty("item_number", "DOC-001");
doc.setProperty("name", "My Document");
doc.setProperty("description", "Description here");

Item result = doc.apply();
if (result.isError())
{
    throw new Exception(result.getErrorString());
}

string docId = result.getID();
```

Uploading a File

```csharp
```

```
// Create file item
Item file = innovator.newItem("File", "add");
file.setProperty("filename", Path.GetFileName(filePath));
file.attachPhysicalFile(filePath);

// Create relationship to document
Item docFile = innovator.newItem("Document File", "add");
docFile.setProperty("related_id", file);

// Add to document
doc.addRelationship(docFile);

Item result = doc.apply();
```

Searching for Items

```csharp
Item search = innovator.newItem("Document", "get");
search.setProperty("item_number", "DOC-001");

Item result = search.apply();
if (!result.isError())
{
    int count = result.getItemCount();
    for (int i = 0; i < count; i++)
    {
        Item item = result.getItemByIndex(i);
        string name = item.getProperty("name");
    }
}
```

Check-out (Lock) Item

```csharp
Item item = innovator.newItem("Document", "lock");
item.setID(documentId);
Item result = item.apply();
```

Check-in (Unlock) Item

```csharp
Item item = innovator.newItem("Document", "unlock");
item.setID(documentId);
Item result = item.apply();
```

4.8 Data Mapping

CATIA to Aras Property Mapping

CATIA Property	Aras Property	ItemType
PartNumber	item_number	Part / Document
Nomenclature	name	Part / Document
Revision	major_rev	Part / Document
Definition	description	Part / Document
Mass	weight	Part
Material	material	Part

File Type Mapping

CATIA Extension	Aras Document Type	Notes
.CATPart	3D Model	Single part
.CATProduct	3D Model	Assembly
.CATDrawing	Drawing	2D drawing
.cgr	Visualization	Graphics rep
.pdf	Drawing	Export
.stp / .step	Exchange	Neutral format

4.9 Development Tasks

Task	Description	Priority	Status
1. Project setup	Create VS solution, add references	High	⏳
2. CATIA connection	Connect to running CATIA instance	High	⏳
3. Reuse Phase 1 code	Copy ArasService, ConfigManager	High	⏳
4. Login form	Adapt from Phase 1	High	⏳
5. Check-in command	Save active document to Aras	High	⏳
6. Check-out command	Open from Aras with lock	High	⏳
7. Get Latest command	Download latest version	High	⏳
8. Search form	Find documents in Aras	Medium	⏳
9. BOM extraction	Read CATProduct structure	Medium	⏳
10. BOM sync	Create/update Aras BOM	Medium	⏳
11. Property sync	Map CATIA ↔ Aras properties	Medium	⏳
12. Settings form	Configure mappings	Low	⏳
13. Error handling	Robust error management	Medium	⏳
14. Logging	Activity logging	Low	⏳
15. Testing	Test all scenarios	High	⏳
16. Documentation	User guide	Medium	⏳

---

5. Phase 4: SAP ERP Integration

5.1 Overview

Integrate Aras Innovator with SAP ERP for bi-directional data exchange of Parts, BOMs, and Documents.

5.2 Integration Scenarios

Direction	Data	Trigger	Priority
Aras → SAP	Released Parts	On Part release	High
Aras → SAP	Released BOM	On BOM release	High
Aras → SAP	Documents	On Document release	Medium
SAP → Aras	Material Master	On request/scheduled	Medium
SAP → Aras	Vendor/Supplier	On request/scheduled	Low

5.3 Architecture

```
┌─────────────┐     ┌─────────────┐     ┌─────────────┐
│             │     │             │     │             │
│    Aras     │◄───►│ Integration │◄───►│   SAP ERP   │
│  Innovator  │ IOM │   Service   │ RFC │             │
│             │     │  (C# .NET)  │OData│             │
└─────────────┘     └─────────────┘     └─────────────┘
```

5.4 SAP Connection Options

Option	Technology	Pros	Cons
SAP .NET Connector (NCo 3.0)	RFC/BAPI	Full functionality, fast	Requires SAP libraries
SAP OData Services	REST	Easy to use, no libraries	Limited functionality
SAP IDocs	File/Message	Good for batch	Asynchronous only

5.5 Key SAP Functions

SAP Function	Purpose
BAPI_MATERIAL_SAVEDATA	Create/Update Material Master
CSAP_MAT_BOM_CREATE	Create BOM in SAP

SAP Function	Purpose
BAPI_DOCUMENT_CREATE	Create Document Info Record (DIR)
RFC_READ_TABLE	Read SAP tables
BAPI_MATERIAL_GET_DETAIL	Get Material details

5.6 Data Mapping: Aras to SAP

Part → Material Master

Aras Property	SAP Field	SAP Table
item_number	MATNR	MARA
name	MAKTX	MAKT
description	MAKTX	MAKT
unit_of_measure	MEINS	MARA
material	MTART	MARA
weight	BRGEW	MARA
make_buy	BESKZ	MARC

BOM → SAP BOM

Aras Property	SAP Field	SAP Table
Parent Part	MATNR	STKO
Child Part	IDNRK	STPO
Quantity	MENGE	STPO
UOM	MEINS	STPO
Position	POSNR	STPO

5.7 Development Tasks

Task	Description	Priority	Status
1. SAP connection	Establish connection to SAP	High	⏳
2. Part release trigger	Aras workflow/method on release	High	⏳
3. Material Master create	Create material in SAP	High	⏳
4. BOM sync	Create BOM in SAP	High	⏳
5. Document sync	Create DIR in SAP	Medium	⏳
6. Error handling	Handle SAP errors	Medium	⏳
7. Logging	Transaction logging	Medium	⏳
8. Status feedback	Show sync status in Aras	Low	⏳
9. Testing	End-to-end testing	High	⏳

---

6. Technical Reference

6.1 Required DLLs

DLL	Source	Purpose
IOM.dll	Aras installation	Aras API
Interop.INFITF.dll	CATIA (generated)	CATIA Infrastructure
Interop.MECMOD.dll	CATIA (generated)	Mechanical Modeler
Interop.PARTITF.dll	CATIA (generated)	Part Interfaces
Interop.ProductStructureTypeLib.dll	CATIA (generated)	Assembly
sapnco.dll	SAP	SAP .NET Connector
sapnco_utils.dll	SAP	SAP Utilities

6.2 CATIA Type Libraries Location

```
C:\Program Files\Dassault Systemes\B32\win_b64\code\bin\
```

Key type libraries:

- InfTypeLib.tlb (Infrastructure)
- MecModTypeLib.tlb (Mechanical Modeler)
- PartTypeLib.tlb (Part)
- ProductStructureTypeLib.tlb (Assembly)
- DrawingTypeLib.tlb (Drawing)

6.3 Generating CATIA Interop Assemblies

Using Visual Studio:

1. Right-click project → Add → COM Reference
2. Select CATIA type libraries
3. Visual Studio generates interop assemblies

Using tlbimp.exe:

```bash
tlbimp "C:\...\InfTypeLib.tlb" /out:Interop.INFITF.dll
```

6.4 Aras IOM.dll Location

```
C:\Program Files (x86)\Aras\Innovator\Client\IOM.dll
```

Or from server:

```
C:\Program Files (x86)\Aras\Innovator\Server\bin\IOM.dll
```

---

7. Development Guidelines

7.1 Coding Standards

- Use C# 8.0+ features
- Follow Microsoft naming conventions
- XML documentation for public methods
- Async/await for long operations
- Proper exception handling
- Logging for debugging

7.2 Error Handling

```csharp
try
{
    // Operation
}
catch (COMException comEx)
{
    // CATIA COM error
    Logger.Error($"CATIA error: {comEx.Message}");
}
catch (Exception ex) when (ex.Message.Contains("Aras"))
{
    // Aras error
    Logger.Error($"Aras error: {ex.Message}");
}
catch (Exception ex)
{
    // General error
    Logger.Error($"Error: {ex.Message}");
    throw;
}
```

7.3 Configuration Storage

Location: `%APPDATA%\ArasCatiaIntegration\config.json`

```json
{
  "ArasServer": "http://localhost/InnovatorServer",
  "ArasDatabase": "InnovatorSolutions",
  "ArasUsername": "admin",
  "RememberPassword": true,
  "LocalWorkspace": "C:\\ArasWorkspace",
  "PropertyMappings": [
    { "CatiaProperty": "PartNumber", "ArasProperty": "item_number" },
    { "CatiaProperty": "Nomenclature", "ArasProperty": "name" }
  ]
}
```

7.4 Logging

Use a simple logging class:

```csharp
public static class Logger
{
    private static string LogFile = Path.Combine(
        Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData),
        "ArasCatiaIntegration", "log.txt");

    public static void Info(string message) => Log("INFO", message);
    public static void Error(string message) => Log("ERROR", message);

    private static void Log(string level, string message)
    {
        string entry = $"{DateTime.Now:yyyy-MM-dd HH:mm:ss} [{level}] {message}";
        File.AppendAllText(LogFile, entry + Environment.NewLine);
    }
}
```

---

8. API Reference

8.1 Aras AML Examples

Get Part by Number

```xml
<Item type="Part" action="get">
  <item_number>PART-001</item_number>
</Item>
```

Create Part with BOM

```xml
<Item type="Part" action="add">
  <item_number>ASSY-001</item_number>
  <name>Assembly</name>
  <Relationships>
    <Item type="Part BOM" action="add">
      <related_id>
        <Item type="Part" action="get">
          <item_number>PART-001</item_number>
        </Item>
      </related_id>
      <quantity>2</quantity>
    </Item>
  </Relationships>
</Item>
```

Lock Item

```xml
<Item type="Document" action="lock" id="ABC123..." />
```

Unlock Item

```xml
<Item type="Document" action="unlock" id="ABC123..." />
```

8.2 CATIA VBA Examples (for reference)

Get Active Part Number

```vba
```

```
Dim partDoc As PartDocument
Set partDoc = CATIA.ActiveDocument
MsgBox partDoc.Part.Parameters.Item("PartNumber").ValueAsString
```

Traverse Assembly

```vba
Sub TraverseAssembly(prod As Product, level As Integer)
    Debug.Print Space(level * 2) & prod.PartNumber
    Dim i As Integer
    For i = 1 To prod.Products.Count
        TraverseAssembly prod.Products.Item(i), level + 1
    Next i
End Sub
```

---

9. Testing Checklist

9.1 CATIA Add-in Tests

Test Case	Steps	Expected Result	Status
Connect to CATIA	Launch add-in with CATIA running	Connection established	⏳
Login to Aras	Enter credentials, click Login	Login successful	⏳
Check-in CATPart	Open part, click Check-in	Document created in Aras	⏳
Check-in CATProduct	Open assembly, click Check-in	Document created in Aras	⏳
Check-out	Select document, click Check-out	File downloaded, locked	⏳
Get Latest	Select document, click Get Latest	File downloaded, not locked	⏳
Search	Enter search term	Results displayed	⏳
BOM Sync	Open assembly, click BOM Sync	BOM created in Aras	⏳
Handle error	Try invalid operation	Error message shown	⏳

9.2 SAP Integration Tests

Test Case	Steps	Expected Result	Status
Connect to SAP	Configure connection	Connection established	⏳
Create Material	Release Part in Aras	Material created in SAP	⏳
Create BOM	Release BOM in Aras	BOM created in SAP	⏳
Update Material	Modify Part, release new rev	Material updated in SAP	⏳
Handle error	Trigger SAP error	Error logged, user notified	⏳

---

10. Troubleshooting

10.1 Common Issues

Issue	Cause	Solution
Cannot connect to CATIA	CATIA not running	Start CATIA first
COM exception	Wrong CATIA version	Regenerate interop assemblies
Aras login fails	Wrong credentials	Check server URL, database, credentials
File upload fails	Network issue	Check vault path and permissions
BOM sync fails	Missing parts	Ensure all parts exist in Aras

10.2 Debug Mode

Set environment variable for verbose logging:

```
ARAS_CATIA_DEBUG=1
```

10.3 Support Resources

Resource	URL
Aras Community	https://community.aras.com
Aras Documentation	https://aras.com/support/documentation
ArasLabs GitHub	https://github.com/ArasLabs
CATIA Documentation	Dassault Systèmes support portal

Appendix A: Claude Code Prompts

Starting Phase 3 Development

```
I'm continuing development of an Aras Innovator integration project.

Context:
- Phase 1 (Windows Send To tool) is complete
- Now building Phase 3: CATIA V5 Add-in
- CATIA version: V5-6R2022 SP6 (COM API)
- Aras: Community Edition 14+
- Framework: .NET 4.7.2, C#

Please review the project guide document and help me:
1. Set up the Visual Studio project structure
2. Create the CATIA connection wrapper class
3. Implement the Check-in command

Reference the Aras_Integration_Project_Guide.md for technical details.
```

Starting Phase 4 Development

```
I'm continuing development of an Aras Innovator integration project.

Context:
- Phase 1 (Windows Send To) is complete
- Phase 3 (CATIA Add-in) is complete
- Now building Phase 4: SAP ERP Integration
- SAP connection method: (NCo 3.0 / OData)
- Framework: .NET 4.7.2, C#

Please review the project guide document and help me:
1. Set up SAP connection
2. Create Material Master from Aras Part
3. Sync BOM to SAP

Reference the Aras_Integration_Project_Guide.md for technical details.
```

Appendix B: File Templates

ConfigManager.cs Template

```
csharp
```

```
using System;
using System.IO;
using Newtonsoft.Json;

namespace ArasCatiaIntegration
{
    public class ConfigManager
    {
        private static readonly string ConfigFolder = Path.Combine(
            Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData),
            "ArasCatiaIntegration");

        private static readonly string ConfigFile = Path.Combine(ConfigFolder, "config.json");

        public AppConfig Config { get; private set; }

        public ConfigManager()
        {
            Load();
        }

        public void Load()
        {
            if (File.Exists(ConfigFile))
            {
                string json = File.ReadAllText(ConfigFile);
                Config = JsonConvert.DeserializeObject<AppConfig>(json);
            }
            else
            {
                Config = new AppConfig();
            }
        }

        public void Save()
        {
            if (!Directory.Exists(ConfigFolder))
                Directory.CreateDirectory(ConfigFolder);

            string json = JsonConvert.SerializeObject(Config, Formatting.Indented);
            File.WriteAllText(ConfigFile, json);
        }
    }
    public class AppConfig
    {
        public string ArasServerUrl { get; set; } = "http://localhost/InnovatorServer";
        public string ArasDatabase { get; set; } = "InnovatorSolutions";
        public string ArasUsername { get; set; } = "admin";
        public string ArasPassword { get; set; } = "";
        public bool RememberPassword { get; set; } = false;
        public string LocalWorkspace { get; set; } = @"C:\ArasWorkspace";
    }
}
```

Document Information

Field	Value
Document Title	Aras Integration Project Guide
Version	1.0
Created Date	December 2024
Last Updated	December 2024
Author	Implementation Team
Purpose	Claude Code development reference

End of Document