Paylocity Employee Change Webhook Integration

Executive Summary

This Azure Functions-based integration provides **automated, real-time synchronization** of employee data changes from **Paylocity** (HR/Payroll platform) to **Microsoft Entra ID** (Azure Active Directory). When employee attributes change in Paylocity—such as job title, department, or reporting manager—this webhook automatically propagates those changes to the organization's identity directory, ensuring data consistency across HR and IT systems.

Business Value

Benefit	Description
Automated Sync	Eliminates manual data entry and reduces human error
Real-Time Updates	Employee changes reflect immediately in directory services
Audit Compliance	All changes logged to Application Insights for compliance and troubleshooting
Dual Environment	Sandbox environment enables safe testing before production deployment

---

Architecture Overview

```
┌──────────────────────┐          ┌──────────────────────┐
┌──────────────────────┐
│                      │          │                      │          │
│
│      PAYLOCITY       │─────────▶│    AZURE FUNCTION    │─────────▶│  MICROSOFT ENTRA
ID  │
│    (HR Platform)     │ Webhook  │   (webhookhandler)   │  Graph   │   (Azure AD)
│
│                      │          │                      │   API    │
│
└──────────────────────┘          └───────────┬──────────┘
└──────────────────────┘
                                              │
                                              │ Telemetry
                                              ▼
                                  ┌──────────────────────┐
                                  │  APPLICATION         │
                                  │  INSIGHTS            │
                                  │  (Logging & Audit)   │
                                  └──────────────────────┘
```

Data Flow

Step-by-Step Process

1. **Webhook Trigger** Paylocity sends an HTTP POST to `/api/webhookhandler` containing `companyId` and `employeeId`
2. **Environment Resolution** System determines Sandbox vs Production based on company ID and loads appropriate credentials
3. **Source Data Fetch** Authenticates to Paylocity API and retrieves complete employee record including supervisor information
4. **Data Transformation** Maps Paylocity fields to Entra ID schema and generates possible User Principal Name (UPN) variations
5. **User Lookup** Searches Entra ID using multiple UPN patterns to handle naming variations
6. **Change Detection** Compares current Entra values against incoming Paylocity data to identify actual changes
7. **Directory Update** Updates modified attributes (job title, department, manager) via Microsoft Graph API
8. **Audit Logging** Records complete change details to Application Insights for compliance tracking

---

Integrated Services

Paylocity API

- **Role**: Source of truth for employee HR data
- **Authentication**: OAuth 2.0 Client Credentials
- **Data Retrieved**: Employee name, job title, department (cost center), supervisor

Microsoft Graph API

- **Role**: Updates user attributes in Entra ID
- **Authentication**: MSAL (Microsoft Authentication Library) with Client Credentials
- **Operations**: User lookup, attribute updates, manager relationship assignment

Azure Application Insights

- **Role**: Centralized logging and monitoring
- **Data Captured**: Employee changes, error events, audit trail with before/after values

---

Environments

Environment	Paylocity Company ID	Azure Tenant	Email Domain
Sandbox	`CATALYST`	`ab558d63-c75b-4100-ae33-9160bbbcfbaa`	`healthplansai.dog`
Production	`163160`	`d39a588a-b5a6-4378-9f5f-f9a0e5484b06`	`catalystsolutions.com`

Synchronized Attributes

Paylocity Field	Entra ID Field	Notes
`jobTitle`	`jobTitle`	Direct mapping
`costCenter1`	`department`	Cost center serves as department
`supervisorEmployeeId`	`manager`	Resolves supervisor to Entra user and sets manager relationship

---

Technical Stack

Component	Technology
Runtime	Azure Functions v2 (Node.js 20.x)
HTTP Client	Axios
Azure Auth	@azure/msal-node
Monitoring	Application Insights SDK
CI/CD	GitHub Actions
Hosting	Azure Function App (Windows)

Deployment Architecture

```
┌─────────────────────────────────────────────────────────────────┐
│                       GitHub Repository                          │
│                       (master branch)                            │
└──────────────────────────┬──────────────────────────────────────┘
                           │
                           │ Push to master
                           ▼
┌─────────────────────────────────────────────────────────────────┐
│                       GitHub Actions                              │
│  • Checkout code                                                  │
│  • Install dependencies                                           │
│  • Build and test                                                 │
│  • Package artifact                                               │
│  • Deploy to Azure                                                │
└──────────────────────────┬──────────────────────────────────────┘
                           │
              ┌────────────┴────────────┐
              ▼                         ▼
┌─────────────────────────┐  ┌─────────────────────────────────┐
│   SANDBOX FUNCTION      │  │   PRODUCTION FUNCTION           │
│ cs-paylocity-emp-       │  │ paylocity-empchange-prod-       │
│ change-func             │  │ westus3-func                    │
└─────────────────────────┘  └─────────────────────────────────┘
```

API Contract

Webhook Endpoint

POST `/api/webhookhandler`

Request Body

```
{
  "companyId": "CATALYST",
  "employeeId": "10546"
}
```

Response Codes

Status	Description
`200`	Success - returns changed fields and environment
`400`	Bad Request - missing required parameters
`404`	User not found in Entra ID
`500`	Internal Server Error - API failure

Success Response Example

```
{
  "status": "success",
  "changedFields": ["jobTitle", "department"],
  "environment": "SANDBOX"
}
```

Monitoring & Observability

Application Insights Events

Event Name: `EmployeeChange`

Properties Tracked:

- Timestamp (ISO 8601)
- Company ID
- Employee ID and full name
- User Principal Name
- Changed fields array
- Detailed change objects (old and new values)
- Environment (PRODUCTION/SANDBOX)
- Error details (if applicable)

Sample KQL Query

```
customEvents
| where name == "EmployeeChange"
| extend companyId = tostring(customDimensions.companyId)
| extend employeeId = tostring(customDimensions.employeeId)
| extend employeeName = tostring(customDimensions.employeeName)
| extend changedFields = tostring(customDimensions.changedFields)
| extend environment = tostring(customDimensions.environment)
| project timestamp, companyId, employeeId, employeeName, changedFields, environment
| order by timestamp desc
```

Security Considerations

- **OAuth 2.0 Client Credentials** flow for all API authentication
- **Environment variables** store all secrets (not in code)
- **HTTPS only** for all API communications
- **Audit logging** provides compliance trail
- **Least privilege** API scopes configured

Project Structure

```
paylocity-employee-change/
├── webhookhandler/
│   ├── index.js                # Main webhook handler
│   └── function.json           # Azure Functions trigger config
├── shared/
│   ├── config.js               # Environment resolution
│   ├── paylocityClient.js      # Paylocity API client
│   ├── graphClient.js          # Microsoft Graph client
│   ├── mapPaylocityToEntra.js  # Data transformation
│   └── logger.js               # Application Insights logging
├── .github/workflows/          # CI/CD pipelines
├── package.json                # Dependencies
└── host.json                   # Azure Functions config
```

Key Dependencies

Package	Version	Purpose
`@azure/msal-node`	^3.8.2	Microsoft authentication
`applicationinsights`	^2.9.8	Telemetry and logging
`axios`	^1.13.2	HTTP client
`dotenv`	^17.2.3	Environment configuration

Important Notes

> ***Warning**: Pushing to the `master` branch triggers automatic deployment to Azure. All changes should be tested in the sandbox environment before production release.*

Summary

This integration serves as a critical bridge between Paylocity HR data and Microsoft Entra ID, ensuring that employee organizational data remains synchronized across systems. The architecture prioritizes reliability through comprehensive error handling, auditability through Application Insights logging, and safety through dual-environment deployment.

Document generated: January 2026