opentext™

OpenText™ Identity Manager
Driver for Epic Implementation Guide

January 2025

Legal Notice

Copyright 2009 - 2024 Open Text.

The only warranties for products and services of Open Text and its affiliates and licensors (“Open Text”) are as may be set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Open Text shall not be liable for technical or editorial errors or omissions contained herein. The information contained herein is subject to change without notice.

Contents

Understanding the Epic EMP Driver **5**

1 Driver Features **7**

Supported Operations 7
Entitlement Support 8
Special Attribute Handling 8

2 Installing the Epic Driver **11**

System Requirements 11
Driver Dependencies 11
Security Recommendations 11
Driver Packages 12
Driver Set Packages 13
User Application Packages 13
Epic Setup 13
- Epic Service Account Setup 13
- Epic ID Type Setup 14
- Web Service Information 14
- Epic Interconnect Username Security Policy Setup 15

Installing the Driver Files 18
Extending Schema for Supporting Custom Attributes 18
Creating the Driver Object 19
- Importing the Current Driver Packages 19
- Activating the Driver 20

Configuring the Epic Driver 20
- Configuration for the Driver Set - Common Settings 20
- Configuration for the Epic driver - Authentication 20
- Driver Parameters 21
- Global Configuration Values 22

A Appendix A - Troubleshooting the Driver **31**

B Appendix B - Primary Manager **33**

C Appendix C - Definitions, Acronyms, and Abbreviations **35**

D Appendix D – Supported Epic EMP Attributes **37**

E Appendix E – Supported Epic SER Attributes **47**

F Appendix F – Special SER Attributes **49**

Epic Provider Address (21000 attributes) 49
Global Configuration Values (GCVs). 50
Single Address 50
Multiple Addresses 50
Facsimile Telephone Number. 51

Understanding the Epic EMP Driver

Epic is an Electronic Medical Records (EMR) management system that maintains medical records and provides providers and patients with access to said medical records. Epic is the leading EMR system in use throughout the healthcare industry. Epic contains PHI and PII and is therefore required to meet HIPAA regulatory requirements.

Driver Concepts

The Identity Manager (IDM) Driver for Epic is a connector that allows for identity life cycle management of Epic EMP (login account) and Epic SER (provider) records.

Since Epic does not have an event system or full database query facility, the driver operates on the Subscriber channel only.

Epic does not utilize groups as part of its security architecture. Instead, Epic utilizes the concept of security templates to assign access within the application. The Epic driver takes advantage of this architecture and allows the assignment of these templates and sub-templates within Epic via the assignment of IDM entitlements. A user may only have a single active template, but the assignment of sub-templates provides for supplemental security access within the Epic system.

Due to user associations, the best practice is to have a separate driver instance for EMP and SER communications.

Data Transfer Between Systems

The Epic driver communicates with Epic using Epic's Web Services.

Thus, an Epic Interconnect Web Server must be licensed and enabled within each customer's deployment Epic environments. Contact the Epic administrator to enable this service.

Publisher Channel

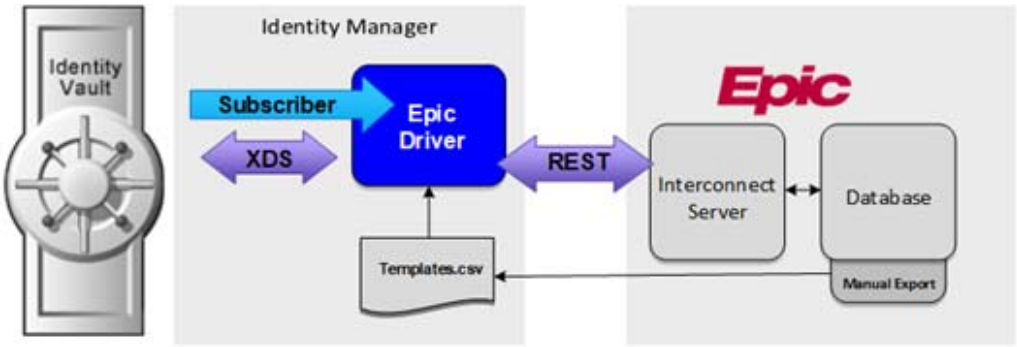
Due to the Epic architecture, the publisher channel is not supported on this driver.

Subscriber Channel

The Epic driver synchronizes the User class and attributes from the Identity Vault to Epic, via the subscriber channel. Attributes synchronized are specific to each implementation but typically contain the CN, Given Name, Surname, Login Disabled, and other user-level attributes.

How the Driver Works

The Epic driver shim is a Java shim that communicates with Epic environments via an Epic Interconnect server and Epic REST web services.



The manually exported CSV files contain the entitlement data (such as Linkable Templates, Sub-Templates, External Identifiers, In Basket Classifications, and Blueprints) that needs to be regularly exported by your Epic Security team and copied to the Identity Manager server.

Standard Use Cases

The following is a list of standard use cases and benefits of the Epic driver:

- Automatic record creation in Epic.
- Automatic updates in Epic on managed records and attributes.
- Mapping of Epic security template assignment to job or business roles in IDM.
- Mapping of Epic security template assignment to business roles or technical roles in Identity Governance.
- Security templates may be assigned via driver policy in the Epic driver.
- Security templates may be assigned via role and resources in IDM Dash.
- Security templates may be assigned via role and resources in Identity Governance.
- Ability to re-evaluate Epic security when roles change within the organization.
- Ability to enforce least level of security within Epic.
- Management of external account assignments within Epic.
- Management of SER record assignments within Epic.

1 Driver Features

This section outlines the features and operations available in the Epic Driver.

- "Supported Operations" on page 7
- "Entitlement Support" on page 8
- "Special Attribute Handling" on page 8

Supported Operations

- Add Operations
  - Add operations are supported for both EMP and SER records
- Modify Operations
  - Add Value is supported for both EMP and SER records
  - Remove Value is supported for EMP records
  - Remove All Values is supported for both EMP and SER records
- Delete Operations
  - Delete operations are supported for EMP records, but severely discouraged by Epic
- Rename Operations
  - Rename operations are not supported by Epic
- Move Operations
  - Move operations are not supported by Epic
- Query Operations
  - The following query operations and classes are supported:
    - EMP class
      - Supports `SystemLoginID`, `InternalID`, `ExternalID`, and `CID`
    - SER class
      - Supports `Identifier` attribute with the format "`userIDType|userID`"
    - Configured entitlement classes
      - Only supported for Code Map Refresh events
      - Query executes against configured CSV files
  - `<query-ex>` operations are not supported by Epic
- Modify Password Operations
  - Modify Password operations are supported for EMP records
- Check Object Password Operations
  - Check Object Password Operations are not supported by Epic

- External Identifiers and Passwords
  - External Identifiers and External Passwords are supported

Entitlement Support

The default driver configuration supports entitlements for linkable templates, sub-templates, in basket classifications, blueprints, and external identifiers in Epic. This is accomplished by exporting, from Epic, the relevant data to individual CSV files.

Sample CSV files and formats for the default entitlements are as follows:

Templates.csv (ID,NAME,DESCRIPTION)

```
T00024,CW NURSE TEMPLATE,CW Nurse
T00076,UTILIZATION MANAGER TEMPLATE,Utilization Manager
```

Subtemplates.csv (ID,NAME)

```
1050003,RADIOLOGY HEART CARE FRONT DESK TEMPLATE BUILDER HE
10801,SBO REV/USAGE SUBTEMPLATE
```

ExternalIdentifiers.csv (ID,NAME)

```
1,Document Imaging
2,Dynamic Imaging
```

InBasketClassifications.csv (ID,NAME)

```
1,Staff
2,WORKSTATION
```

Blueprints.csv (ID,NAME)

```
B10000,AMBULATORY PHYSICIAN
B10057,ANESTHESIOLOGIST
```

When a code map refresh is issued to the driver, the query results in the driver shim reading the previously exported CSV file and returning the entitlement values to IDM.

Special Attribute Handling

Login Disabled: When mapping **Login Disabled** in the Identity Vault to **IsActive** in Epic the value must be reversed for the desired outcome. When **Login Disabled** is set to `true` in the Identity Vault the matching result in Epic for **IsActive** would be `false`. If the attribute us synced straight through without changing the value, the result will be the opposite of what is desired.

Transform isActive - Login Disabled Mapping

When Login Disabled is changing to true, set isActive to false. Otherwise, set isActive to true.

Conditions

Condition Group 1

```
    if class name equal "User"
And if operation attribute 'IsActive' changing
```

Actions

```
if
    if operation attribute 'IsActive' changing to "true"
then
    strip operation attribute("IsActive")
    add destination attribute value("IsActive", "false")
else
    strip operation attribute("IsActive")
    add destination attribute value("IsActive", "true")
```

CustomUserDictionaries: This attribute has been deprecated in Epic and is no longer available since Epic version 2014.

2 Installing the Epic Driver

This section discusses and outlines the installation process of the Epic Driver.

- "System Requirements" on page 11
- "Driver Dependencies" on page 11
- "Security Recommendations" on page 11
- "Driver Packages" on page 12
- "Driver Set Packages" on page 13
- "User Application Packages" on page 13
- "Epic Setup" on page 13
- "Installing the Driver Files" on page 18
- "Extending Schema for Supporting Custom Attributes" on page 18
- "Creating the Driver Object" on page 19
- "Configuring the Epic Driver" on page 20

System Requirements

The Epic driver requires a minimum version of Identity Manager 4.7. If entitlements are being used, then IDM 4.7 Advanced Edition or newer is required.

The Epic driver will run on the IDM engine or on a remote loader.

Driver Dependencies

- URL of each target Epic environment's Interconnect Server endpoint.
- Epic Service Account user ID and password, configured with the appropriate Epic permissions.
- CSV files containing the entitlement data. These should be provided by the Epic security team and regularly updated on the IDM server.

Security Recommendations

- The Epic driver must have the ability to read objects and attributes listed in the driver subscriber filter in addition to standard Identity Manager driver security requirements.

  For more information about object synchronization, see Synchronizing Objects (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/b3uv74q.html) in the *OpenText™ Identity Manager CE 24.4 (v4.10) Driver Administration Guide (https://www.netiq.com/documentation/identity-manager-4.10/driver_admin/data/bktitle.html)*.

- Epic's Web Services are accessible over HTTPS. The Epic driver must be able to connect to the Epic Interconnect server over this https connection. Ensure that any firewall rules are updated to allow the Epic driver to communicate with Epic (TCP port 443 unless otherwise configured in the Epic implementation).
- Audit User in Epic – The identifier of the record creating the new User record. Epic recommends that this field be left blank, although some implementations may require a value.
- ? Epic Client ID – Starting with the February 2019 Epic build, all API calls must have a Client ID. The Epic driver uses the http header option.

  For more information, see "Activating the Driver" on page 20.
- Trust all Certs – There are security risks associated with utilizing this functionality as it can open the system to potential MIM attacks. Therefore, though it is not recommended, if the Epic system is utilizing a self-signed certificate, the "Trust All Certs" driver configuration may be enabled.

Driver Packages

The following driver packages are utilized by the Epic driver when configured for EMP support:

Package Name	Min Version	Description
CPRS Common	1.0	Provides low-level functionality like skipping schema mapping for query results for certain CPRS queries.
Custom Entitlements	1.0	Provides support for making the custom entitlements available for RBPM (IDM Catalog), by adding them to the entitlement configuration object during driver start.
Epic Base	3.0	Epic driver base package.
Epic EMP Default Configuration	3.0	Default EMP configuration parameters and policies. Only use when the driver is managing EMP records.
Epic EMP Entitlements	3.0	Support for EMP entitlements. Only use when the driver is managing EMP records.
Password Synchronization Common	2.1	Support for password synchronization. Only use when the driver is managing passwords.

The following driver packages are utilized by the Epic driver when configured for SER support:

Package Name	Min Version	Description
CPRS Common	1.0	Provides low-level functionality like skipping schema mapping for query results for certain CPRS queries.
Custom Entitlements	1.0	Provides support for making the custom entitlements available for RBPM (IDM Catalog), by adding them to the entitlement configuration object during driver start.
Epic Base	3.0	Epic driver base package.

Package Name	Min Version	Description
Epic SER Default Configuration	3.0	Default SER configuration parameters and policies. Only use when the driver is managing SER records.
Epic SER Entitlements	3.0	Support for SER entitlements. Only use when the driver is managing SER records.

Driver Set Packages

The following driver set packages are utilized by the Epic driver:

Package Name	Min Version	Description
Advanced Java Class (NOVLLIBAJC)	2.2	ECMAscript used by the driver policies.
Common Settings Advanced Edition	2.1	Configuration used by the driver policies.

User Application Packages

The following driver set packages are utilized by the Epic driver:

Package Name	Min Version	Description
User Application Epic EMP Resources	3.0	EMP resources and workflows.

Epic Setup

- "Epic Service Account Setup" on page 13
- "Epic ID Type Setup" on page 14
- "Web Service Information" on page 14
- "Epic Interconnect Username Security Policy Setup" on page 15

Epic Service Account Setup

The Epic driver requires an Epic user for username token authentication (Basic Auth) to the Epic database. This user is used by Interconnect to authenticate the driver's web service calls. Communicate to your Interconnect Admin the ID (EMP .1) and Password (EMP 40) of this user after you create it. Note that both the ID and Password fields are case-sensitive.

- This should be an active foreground user and should have a password set to not expire.

- It is recommended to create new security classes or to use static security classes (see relevant EMP security points below), as changes to this user's security impacts the Epic driver's functionality, and then add this security class to this user record.
- The security points for the user must be set using the default security class for a given Epic application. Note that the user's security will not inherit security class overrides by service area or location.

Epic ID Type Setup

Epic requires each customer to have a unique Epic identifier to hold the SER record ID for the Epic Driver, similar to an NPI identifier. If a desired type does not already exist, a new Epic type will need to be created.

Ensure that a HL7 assigning authority is configured in the settings for this ID type such that the Type.text is available when reading Provider records.

Web Service Information

The Epic driver is scoped for the following web services:

Web Service Name	Web Service Category	Web Service Class
ActivateUser	Security	PersonnelManagement
CreateUser (2014)	Security	PersonnelManagement
DeleteUser	Security	PersonnelManagement
GetImportDataLog	Core	DataUtility
GetInternalIdentifier	Common	Utility
GetUserPagerID	Security	PersonnelManagement
ImportData (2019)	Core	DataUtility
InactivateUser	Security	PersonnelManagement
Practitioner.Read (R4)	FHIR	R4
Practitioner.Search (R4)	FHIR	R4
PractitionerRole.Read (R4)	FHIR	R4
PractitionerRole.Search (R4)	FHIR	R4
SetReportSelectionCriteria	Security	PersonnelManagement
SerUserExternalPasswords	Security	PersonnelManagement
SetUserPagerID	Security	PersonnelManagement
SetUserPassword	Security	PersonnelManagement
UpdateAuthorizedServiceAreas	Security	PersonnelManagement
UpdateBIDefaultUser	Security	PersonnelManagement

Web Service Name	Web Service Category	Web Service Class
UpdateCommunityUser	Security	PersonnelManagement
UpdateFacilityDepartmentList	Security	PersonnelManagement
UpdateLoginDepartments	Security	PersonnelManagement
UpdateUser (2014)	Security	PersonnelManagement
UpdateUserDemographics	Common	User
UpdateUserGroups	Security	PersonnelManagement
ViewAuthorizedServiceAreas	Security	PersonnelManagement
ViewBIDefaultUser	Security	PersonnelManagement
ViewCommunityUser	Security	PersonnelManagement
ViewCurrentReportSelectionCriteria	Security	PersonnelManagement
ViewFacilityDepartmentList	Security	PersonnelManagement
ViewLoginDepartments	Security	PersonnelManagement
ViewUser (2014)	Security	PersonnelManagement
ViewUserGroups	Security	PersonnelManagement

Epic Interconnect Username Security Policy Setup

Epic Interconnect needs to be set up to allow the Epic driver to access data in Epic.

Recommended Location	Server: Background Instance: Web Service Host
Role	General Web Service Host
Cache Listeners	N/A
Business Services	Integration-specific web services ◆ DataUtility (Core) (WSC) ◆ PersonnelManagement (Security) (WSC) ◆ R4 (FHIR) (WSC) ◆ Utility (Common) (WSC) ◆ User (Common) (WSC) See Web Service Information for the full list of web services user by the Epic driver.

Security Policy	Policy Name: EpicDriverBasicAuth ◆ Services: Application services from above ◆ Authentication: Username tokens (use the EMP ID from the Epic Service Account Setup section) ◆ Encryption: TLS ◆ Bindings: Rest

Reverse Proxy Pattern(s)

Be sure you have configured your reverse proxy (see the Configure ARR Manually section in Epic's Interconnect Setup & Support Guide).

Create a rewrite rule for each pattern below using Wildcards instead of the default Regular Expressions:

- api/fhir/*
- api/epic/2011/Common/User/UPDATEUSERDEMOGRAPHICS/*
- api/epic/2010/Common/Utility/GETINTERNALIDENTIFIER/*
- api/epic/2016/Core/DataUtility/GetImportDataLog/*
- api/epic/2019/Core/DataUtility/ImportData/*
- api/epic/2012/Security/PersonnelManagement/ActivateUser/*
- api/epic/2012/Security/PersonnelManagement/DeleteUser/*
- api/epic/2012/Security/PersonnelManagement/InactivateUser/*
- api/epic/2012/Security/PersonnelManagement/SetUserExternalPasswords/*
- api/epic/2012/Security/PersonnelManagement/SetUserPassword/*
- api/epic/2014/Security/PersonnelManagement/CreateUser/*
- api/epic/2014/Security/PersonnelManagement/UpdateUser/*
- api/epic/2014/Security/PersonnelManagement/ViewUser*
- api/epic/2015/Security/PersonnelManagement/SetReportSelectionCriteria*
- api/epic/2015/Security/PersonnelManagement/ViewCurrentReportSelectionCriteria/*
- api/epic/2016/Security/PersonnelManagement/UpdateCommunityUser/*
- api/epic/2016/Security/PersonnelManagement/UpdateLoginDepartments/*
- api/epic/2016/Security/PersonnelManagement/UpdateUserGroups/*
- api/epic/2016/Security/PersonnelManagement/ViewCommunityUser/*
- api/epic/2016/Security/PersonnelManagement/ViewLoginDepartments/*
- api/epic/2016/Security/PersonnelManagement/ViewUserGroups/*
- api/epic/2017/Security/PersonnelManagement/UpdateAuthorizedServiceAreas/*
- api/epic/2017/Security/PersonnelManagement/UpdateFacilityDepartmentList/*
- api/epic/2017/Security/PersonnelManagement/GetUserPagerID/*
- api/epic/2017/Security/PersonnelManagement/SetUserPagerID/*
- api/epic/2017/Security/PersonnelManagement/ViewAuthorizedServiceAreas/*
- api/epic/2017/Security/PersonnelManagement/ViewFacilityDepartmentList/*
- api/epic/2017/Security/PersonnelManagement/UpdateBIDefaultUser/*
- api/epic/2017/Security/PersonnelManagement/ViewBIDefaultUser/*

Each rule should use Wildcards and rewrite to: https://[hostname]/[instance]/{R:0}

NOTE: : These Reverse Proxy rules are related to Microsoft's Application Request Routing (ARR) for IIS. If you use another reverse proxy solution, check with your Client Systems Web and Service Server TS for assistance.

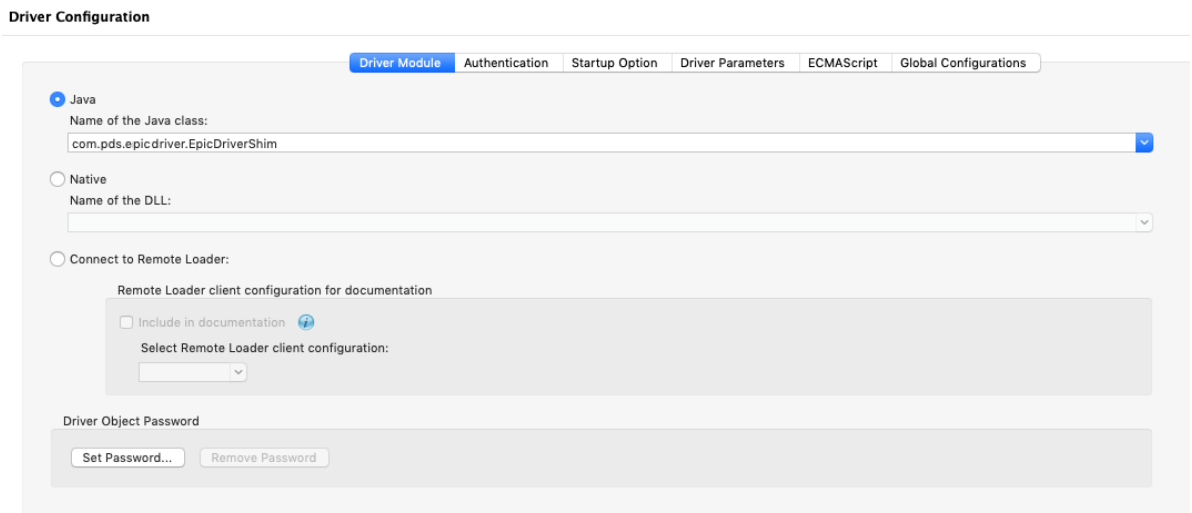
Installing the Driver Files

To install the driver files you must perform one of the following actions based on your platform:

- **Linux**:
  1. Navigate to the extracted driver zip folder > **linux**.
  2. Install the `netiq-DXMLEpic.rpm` in your driver installation directory by running the following command in a terminal window.

     `rpm -Ivh <rpm file path>/netiq-DXMLEpic.rpm`
  3. Restart Identity Vault.
- **Windows**:
  1. Navigate to the extracted driver zip folder > **windows**.
  2. Copy the `EpicDriverShim.jar` file into the `/opt/ novell/eDirectory/lib/dirxml/classes` directory, or `\Novell\RemoteLoader\lib` if the driver is installed with the Remote Loader.
  3. Restart Identity Vault.

When creating the Designer driver object, set the name of the Java class to `com.pds.epicdriver.EpicDriverShim`, as shown in the following image:



Extending Schema for Supporting Custom Attributes

For full support of driver functionality, the driver uses schema not available in the base eDirectory schema definition. A schema file containing the necessary attributes has been provided. The classes and attributes defined in this schema file are as follows:

- Classes
  - pdsEpicUser – Auxiliary class
- Attributes (Optional Attributes of pdsEpicUser class)
  - pdsEmployeeEndDate – Time, Single Valued, Sync Immediate
  - pdsEmployeeEndDateString – Case Ignore String, Single Valued, Sync Immediate

- pdsEmployeeStartDate – Time, Single Valued, Sync Immediate
- pdsEmployeeStartDateString – Case Ignore String, Single Valued, Sync Immediate
- pdsEpicLinkSER – Boolean, Single Valued, Sync Immediate
- pdsProviderAddress – Case Ignore String, Sync Immediate

You can upload new attributes through the Identity Manager to extend the schema. The following steps explain the procedure to extend the schema:

1. Navigate to the extracted driver zip folder **> schema.**
2. Copy the `.sch` file to the system where Identity Manager is installed.
3. Execute the following `ndssch` command.

   ```
   ndssch [-h hostname[:port]] [-t tree_name] [-d] admin_FDN schemafile
   [schema_description]
   ```

   For example, `ndssch -h 10.71.131.123:524 -t SLES12SP3_Quality_131123_TREE -d admin.sa.system /root/schema/pdsEpicUser.sch`
4. The log file is created in the default location, `/root/schema.log` for troubleshooting.

Creating the Driver Object

To create the Epic driver object in Designer, the previously listed driver packages must first be installed in the Designer project's Package Catalog. When creating the driver object, these packages must be added to the driver object, the driver set, and the User Application driver object, and then configured for the target environment.

Importing the Current Driver Packages

To import the driver packages into Designer, use the following steps:

1. Open Designer.
2. In the Outline view, right-click the **Package Catalog**.
3. Click **Import Package**.
4. Browse to the location where the Epic Base, Epic EMP Default Configuration, Epic EMP Entitlements, Epic SER Default Configuration, Epic SER Entitlements packages were downloaded.
5. Select the Epic Base package, and then the Epic (EMP or SER) Default Configuration and Epic (EMP or SER) Entitlements respective packages, for the Epic driver.
6. Click **Select All** to import all of the packages displayed on the screen.
7. Click **OK** to import the selected packages, then click **OK** on the successfully imported packages message.
8. After the packages are imported the driver object may be created and configured for the target environment.

Activating the Driver

The Epic driver is activated by loading the Epic Driver OpenText license key.

Note that the driver must be activated within 90 days of installation, else the driver stops working and an activation error will be displayed in the driver trace file.

Additionally, the driver must also be registered with Epic for use in the customer's Epic environments. This is done in 1 of 3 manners:

- The customer's App Orchard Point Person (AOPP) searches the Epic Connection Hub for the application name "IdentitySync – IAM Integration Module for Micro Focus IDM".
- The customer's App Orchard Point Person (AOPP) contacts their Epic App Orchard TS and provides them with the application name "IdentitySync – IAM Integration Module for Micro Focus IDM".
- The customer contacts their OpenText Account Executive who could have the driver registered on the customer's behalf.

Configuring the Epic Driver

This section discusses and outlines the configuration to be set to use the Epic Driver.

When configuring the driver, a decision must be made as to whether the driver will provision EMP accounts or SER records. As a driver may only maintain a single association per user, both EMP and SER may not be configured on the same driver instance as that would require two associations for the same instance. The “Epic Base” package is applied to the driver regardless of whether it is intended to provision EMP accounts or SER records. However, the primary functionality of the instance is determined by the application of either the “Epic EMP Default Configuration” package or the “Epic SER Default Configuration” package and the matching entitlement package.

Configuration for the Driver Set - Common Settings

Parameter	Description
User Container	DN of base container for user objects.

Configuration for the Epic driver - Authentication

Parameter	Description
Authentication ID	ID of the Epic Service Account. The `format is emp$<ID>.`
Set Password	Password of the Epic Service Account.
Connection Information	URL of the Epic Interconnect Server endpoint.

Driver Parameters

The following driver parameters are set for the Epic EMP driver

Driver Options

Parameter	Description
Epic Version	Version of the Epic implementation. This determines the Epic schema supported by the driver.
Epic Environment	Select Production or Non-Production.
ID Types	A list of custom ID types (Item 20700/20701) to support. This is implementation specific and generally not needed. When an ID type is added here it will show in the Schema as `IDType-<value>`. The value listed here comes from the `ID type descriptor` under **Names in Other Systems** in the ID type definition in Epic. **NOTE:** In the ID Type definition under the ID Rules tab in Epic, the **Method** must be **user entered** or **system generated**.
Trust All Certs	Select Yes or No. This configures the driver whether to trust all certs when establishing the https connection to the Epic SOAP endpoint (there are security risks associated with utilizing this functionality as it can open the system to potential MIM attacks).
Enable Data Courier Log File	Select Yes or No.
Force Contact on all Updates	Select Yes or No. If set to yes, Epic will create a new contact record on every update from the driver.
Audit User	External ID of the user in Epic that will be identified as the creator of a new records. Generally, this is left blank.
Audit User Password	Audit user's password.
Audit User IDType	ID type of the audit user's ID. Generally, this is `External`.
SER Configuration	Select Show or Hide. Select Show when the Epic driver is to be used for SER record management.
Custom Attribute for SER ID in Epic	SER record identifier INI Descriptor. For example, PDS. Epic requires each customer to have a unique Epic identifier to hold the SER record ID, similar to an NPI identifier. If a desired type does not already exist, a new Epic type will need to be created.
Custom Attribute Type for SER ID in Epic	SER record identifier INI # for the above type. For example, 200.
SER Blueprint Override	Default value (1) allows overriding assigned Blueprints in Epic.

Driver Options (EMP Specific)

These driver options are GCV references to the “Epic EMP Default Configuration” Global Configuration Values. Refer to the package “Readme” for instructions on how to install these parameters.

Parameter	Description
Epic Date Format	Format pattern for dates, as configured in Epic environment. For example, "MM-dd-yyyy" or "yyyy.MM.dd"
Custom Epic ID Types	List of custom ID types to support (Example value: "POS_VENDOR_ID"). This is implementation specific and generally isn't needed.
Matching Attribute	Identity Vault (eDirectory) attribute to use when matching Epic records. For example, “CN” or “workforceID”
Matching ID Type	Epic ID Type to use when searching for Epic records. For Example, “SystemLogin”

Driver Options (SER Specific)

These driver options are GCV references to the “Epic SER Default Configuration” Global Configuration Values. Refer to the package “Readme” for instructions on how to install these parameters

Parameter	Description
SER Configuration	Select Show or Hide. Select Show when the Epic driver is to be used for SER record management.
Custom Attribute for SER ID in Epic	SER record identifier INI Descriptor. For example, PDS. Epic requires each customer to have a unique Epic identifier to hold the SER record ID, similar to an NPI identifier. If a desired type does not already exist, a new Epic type will need to be created.
Custom Attribute Type for SER ID in Epic	SER record identifier INI # for the above type. For example, 200.
SER Blueprint Override	Default value (1) allows overriding assigned Blueprints in Epic.

Global Configuration Values

Global configuration values (GCVs) are values that can be used to control policy functionality. GCVs are defined on the driver or on the driver set. Driver set GCVs can be used by all drivers in the driver set while driver GCVs can only be used by the driver on which they are defined

Driver GCV Configuration

This section discusses and outlines the GCV configuration of the Epic Driver.

Custom Entitlements

Parameter	Description
List of Custom Entitlements	List of entitlement names that the driver will process during a CodeMapRefresh request. For EMP record management, the default entitlements supported by the driver are `AppliedTemplate`, `AvailableTemplate`, `DefaultTemplate`, `ExternalIdentifier`, `InBasketClassification`, `Subtemplate`, and `UserAccount`. For SER record management, the default entitlements supported by the driver are `Blueprint` and `SERRecord`.

Common Password Settings

Parameter	Description
Set default password if not available	Select Yes to set a default password if password element is not available in a user add event. This controls how the `*-cp-DefaultPassword` policy executes.

EMP Default Configuration

Account Activation and Deactivation Activities

Parameter	Description
Action to take when ‘User Account’ is deleted in IDV	Select `Inactivate Account` or `Block Account`.
Value for IsBlocked	Select True or False. When `Inactivate Account` is selected above, this tells policies to also set the Epic record’s `IsBlocked` value to `True` or `False`.
Text to send as ‘Block Comment’	When True is selected above, this value is included to document the reason why the record was blocked.
Text to send as ‘Block Comment’	When `Block Account` is selected above, this value is included to document the reason why the record was blocked.
Value for IsActive	Select True or False. When Block Account is selected above, this tells policies to also set the Epic record’s `IsActive` value to True or False.
Synchronize ‘Login Disabled’ attribute	Select whether the changes made to the 'Login Disabled' attribute in IDVault should be synced.
Value for IsBlocked	Select True or False. When Inactivate Account is selected above, this tells policies to also set the Epic record’s IsBlocked value to True or False.

Parameter	Description
Text to send as ‘Block Comment’	When True is selected above, this value is included to document the reason why the record was blocked.
Text to send as ‘Block Comment’	When Block Account is selected above, this value is included to document the reason why the record was blocked.
Value for IsActive	Select True or False. When Block Account is selected above, this tells policies to also set the Epic record’s IsActive value to True or False.
Synchronize Account Lockout	Select Yes or No. This tells policies whether to the set the Epic record’s `IsBlocked` status when the IDV user account is locked by intruder detection.

Account Naming Activities

Parameter	Description
Should Epic generate the ‘UserInternalID’ value	Select Yes or No. When Yes is selected, the IDV attribute that is mapped to UserInternalID in the Schema Mapping policy will be used for a new Epic record's UserInternalID field. **NOTE:** If Yes is selected and the mapped IDV user attribute contains no value, then the driver will return an error when attempting to create a new Epic record. When No is selected, Epic will auto generate the UserInternalID value when creating a new Epic record.
Custom Epic ID Types	List of custom ID types to support (Example value: "POS_VENDOR_ID"). This is implementation specific and generally isn't needed.
Matching Attribute	Identity Vault (eDirectory) attribute to use when matching Epic records. For example, “CN” or “workforceID”.
Matching ID Type	Epic ID Type to use when searching for Epic records. For Example, “SystemLogin”.

SER Record Integration

Parameter	Description
Attribute that holds the SER ID for providers	IDV attribute that holds the SER record ID for a linked SER record. Default is CN.
Attribute MPI ID type that links the SER ID for providers	What is the MPI ID type of the SER ID link attribute.
Attribute that triggers the linking of EMP to SER	The boolean attribute that triggers the linking of an SER record to an EMP record.

Start and End Date Attributes

Parameter	Description
Epic Date Format	Format pattern for dates, as configured in Epic environment. For example, "MM-dd-yyyy" or "yyyy.MM.dd".
Start Date Attribute	The attribute in eDirectory that holds the start date.
Is the start date attribute a string or timestamp?	Is the start date a string or timestamp (eDirectory attribute type “Time”).
Start Date Format	Format pattern for dates, as expected in the source attribute. For example, "MM-dd-yyyy" or "yyyy.MM.dd".
End Date Attribute	The attribute in eDirectory that holds the end date.
Is the end date attribute a string or timestamp?	Is the end date a string or timestamp (eDirectory attribute type “Time”).
End Date Format	Format pattern for dates, as expected in the source attribute. For example, "MM-dd-yyyy" or "yyyy.MM.dd".

EMP Entitlements

Parameter	Description
Enable UserAccount Entitlement	Select `Yes` or `No`. Entitlements act like an ON/OFF switch to control account access. When the driver is enabled for entitlements, accounts are only created or disabled when the account entitlement is granted to or revoked from user accounts. Entitlements are granted and revoked by entitlement agents. Three entitlement agents ship with Identity Manager: ◆ **Role-Based Entitlements (RBE)**: RBE is ideal for simple automation. For example, when a user is added to the HR system, the user is automatically granted accounts in other systems ◆ **Workflow**: Workflow is ideal for approvals. For example, when a user is added to the HR system, the manager must approve the accounts for the user. ◆ **Roles Based Provisioning Module (RBPM)**: RBPM is ideal for true and full-featured roles based provisioning. For example, when a user is added to the Accounting role, the user automatically receives all accounts associated with the accounting role. If **Yes** is selected, one of these entitlement agents must be installed and configured for the driver to create and delete accounts. For more information, see the OpenText™ Identity Manager Entitlements Guide (https://www.netiq.com/documentation/identity-manager-4.10/entitlements/data/identity-manager-entitlements.html). **NOTE:** Synchronization of Login `Disabled` is independent of this setting.
Action to take when 'UserAccount' entitlements is revoked	Select **Inactivate Account** or **Block Account**. This is the action to take when the `UserAccount` entitlement is revoked from a user account. **Inactivate Account** prohibits record assignments in Epic. **Block Account** prevents user login in Epic.
Text to send as 'Block Comment'	When **Block Account** is selected above, this value is included to document the reason why the record was blocked.
Should the account also be deactivated	Select **Yes** or **No**. When Yes is selected and **Block Account** is selected above, policies also **Inactivate Account**.
Enable External Identifier Entitlement	Select **Yes** or **No**. Enables the management of External Identifiers via entitlements.
Enable 'InBasketClassification' Entitlement	Select **Yes** or **No**. Enables the management of In Basket Classifications via entitlements.
Enable 'Subtemplate' Entitlement	Select **Yes** or **No**. Enables the management of Sub-Templates via entitlements.
Enable 'AvailableTemplate' Entitlement	Select **Yes** or **No**. Enables the management of Available Template via entitlements.

Parameter	Description
Enable ‘DefaultTemplate’ Entitlement	Select **Yes** or **No**. Enables the management of Default Templates via entitlements.
Enable ‘AppliedTemplate’ Entitlement	Select **Yes** or **No**. Enables the management of Applied Templates via entitlements. It is recommended that this value is set to "No" as updating Applied Templates to currently logged in users is not supported in Epic.

SER Default Configuration

SER ID Configuration – Required Values

Parameter	Description
Identity Vault attribute to map to SER ID	Select Inactivate Account or Block Account.
Custom Attribute for SER ID in Epic	SER record identifier INI Descriptor. For example, PDS. Epic requires each customer to have a unique Epic identifier to hold the SER record ID, similar to an NPI identifier. If a desired type does not already exist, a new Epic type will need to be created.
Custom Attribute Type for SER ID in Epic	SER record identifier # for the above type. For example, 200.
Matching Attribute for SER Record	IDV attribute to use for matching operations.
SER Blueprint Override	Enter 0 when a Blueprint is added, and it should not override a pre-existing blueprint. Enter 1 when a Blueprint is added, and it should override a pre-existing blueprint.
Attribute that triggers the linking of EMP to SER	The boolean attribute that triggers the linking of an SER record to an EMP record.

SER ID Configuration – Provider Address Mapping (Epic 21000 Attributes)

Parameter	Description
Do providers have a single location or multiple locations?	If providers work only at a single location or if the provider address is a JSON formatted attribute, select "Single." If providers have multiple facility addresses associated with their records, select "Multiple." The attributes listed here MUST BE ENABLED in the driver filter as "notify" attributes on the Subscriber channel. DO NOT map these attributes in the driver's Schema Map.
Multiple Addresses – Must be multi-valued string stored in JSON format – Ensure attribute is in the driver filter as ‘notify’ on Subscriber Channel	

Parameter	Description
Provider Address Attribute	The name of the attribute that contains the provider address. This attribute must be a multi-valued attribute and the values must be stored in JSON string format.
Single Address – Ensure attributes are in driver filter as ‘notify’ on Subscriber Channel	
Provider Street Address Attribute	The name of the attribute that contains the provider street address (Epic SER 21010).
Provider Street Address (2) Attribute	The name of the attribute that contains the provider street address (2) (Epic SER 21020).
Provider Street Address (3) Attribute	The name of the attribute that contains the provider street address (3) (Epic SER 21030).
Provider City Attribute	The name of the attribute that contains the provider city (Epic SER 21040).
Provider State Attribute	The name of the attribute that contains the provider state (Epic SER 21050).
Provider Zip Code Attribute	The name of the attribute that contains the provider zip code (Epic SER 21060).
Provider Country Attribute	The name of the attribute that contains the provider country (Epic SER 21080).
Provider Phone Attribute	The name of the attribute that contains the telephone number (Epic SER 21100).
Provider Fax Attribute	The name of the attribute that contains the facsimile (fax) number (Epic SER 21110).
Provider Email Attribute	The name of the attribute that contains the provider's email address (Epic SER 21130).

Epic SER Entitlements Configuration

Parameter	Description
Use SER Account Entitlement	Select **true** or **false**. Entitlements act like an ON/OFF switch to control account access. When the driver is enabled for entitlements, accounts are only created or disabled when the account entitlement is granted to or revoked from user accounts. Entitlements are granted and revoked by entitlement agents. Three entitlement agents ship with Identity Manager: ◆ **Role-Based Entitlements (RBE)**: RBE is ideal for simple automation. For example, when a user is added to the HR system, the user is automatically granted accounts in other systems ◆ **Workflow**: Workflow is ideal for approvals. For example, when a user is added to the HR system, the manager must approve the accounts for the user. ◆ **Roles Based Provisioning Module (RBPM)**: RBPM is ideal for true and full-featured roles based provisioning. For example, when a user is added to the Accounting role, the user automatically receives all accounts associated with the accounting role. When **true** is selected, one of these entitlement agents must be installed and configured for the driver to create and delete accounts For more information, see the OpenText™ Identity Manager Entitlements Guide (https://www.netiq.com/documentation/identity-manager-4.10/entitlements/data/identity-manager-entitlements.html).
Use Blueprint Entitlement	Select true or false. Enables the management of Blueprints via entitlements.
Blueprint Required for User Create	Select Yes or No. When Yes is selected and Use Blueprint Entitlement is true, user add operations will be vetoed if a Blueprint entitlement is not present.

A Appendix A - Troubleshooting the Driver

Refer to the following sections if you are experiencing an issue with the Epic Driver.

Driver is not starting or connecting to Epic

- Validate that the configured Epic Interconnect Server URL is correct
- Validate that the IDM server's firewall ports are open to allow access to Epic
- Validate that the Authentication ID is in the format `EMP:<ID>` (i.e. `EMP:BSMITH`)
- Ensure that the driver is fully activated. For more information, see "Activating the Driver" on page 20.

User update is failing

If a user record is locked in Epic (i.e. currently open in the Epic Hyperspace application), the driver will not be able to update the user. There is an error reported in the driver trace.

Error "com.sun.xml.internal.ws.client.ClientTransportException: The server sent HTTP status code 500: System.ServiceModel.ServiceActivationException" in trace

- This issue can occur occasionally when there are scarce resources in Epic Interconnect (i.e., memory or CPU)
- This issue can occur in particular instances on certain versions of Epic Interconnect
- Turn on driver trace level 5, duplicate issue, then provide the Web Service details (specifically the full URL of the endpoint that was being called when the error occurred) to your Epic support representative

Error "INVALID-CLIENT-ID details: Provided client ID is invalid" in trace

Ensure that the driver is fully activated. For more information, see "Activating the Driver" on page 20.

Exception "java.lang.NoSuchMethodException" in trace

- This exception will be displayed in the trace when attempting to sync an attribute that isn't supported by Epic
- Review the description of the message in the trace to help isolate the specific attribute(s)
- Review the driver Filter policy and ensure that the specific attribute(s) is not set to "Synchronize"

B Appendix B - Primary Manager

There are certain conditions that must be met for a user to be assigned as a manager. These conditions are listed below:

- The Epic manager record's ID must be present in the Users Managers List.
- The target Epic manager must have an `InBasketClassification` with "`In Basket`" security point "`14-Manage Clinic`" or "`16-Trusted Manager`".
- If the value of the user's manager attribute has an association ref (dn reference) the association value is used, otherwise the value is used.

  ```
  <value association-ref="113" timestamp="1530022566#6"
  type="dn">\IDV47_TREE\data\users\Tester2</value>
  ```

  or

  ```
  <value timestamp="1530022566#6" type="dn">FAMMD</value>
  ```

NOTE: The value used must match the `Epic ID` of the target Epic manager.

C Appendix C - Definitions, Acronyms, and Abbreviations

The following key terms are used in conjunction with the Epic driver:

- Contact - point and time version of the EMP record
- Data Courier - Data courier logs contain the ID of records that are created or modified and are used to trigger the target system to retrieve the respective updates from another Epic environment
- EMR – Electronic Medical Records
- EMP Record – User Login Record in the Epic system
- External ID - Value of a user name external to the Epic system
- HIPAA - Health Information Portability and Accountability Act
- IDM - Identity Manager
- Item – Attribute of an EMP record
- Linked (Linkable) Template – Grouping of items to be applied to EMP record (often security related)
- PII – Personally Identifiable Information
- PHI – Personal Healthcare Information
- SER - Schedulable Epic Resource

D Appendix D – Supported Epic EMP Attributes

The following Epic EMP attributes are supported by the Epic Driver:

Address

User’s address.

- Type: Address[]
- Example: { "City": "String", "Country": "String", “County”: “String”, “District”: “String”, “HouseNumber”: “String”. “Lines”: [“String”], “State”: “String”, “ZipCode”: “String” }

AuthenticationConfigurationID

The authentication configuration to use for this user.

- Type: IDType
- Example: { "ID": "String", "Type": "String" }

AuthorizedServiceAreas

A list of the authorized service areas.

- Type: IndexedRecordIn[]
- Example: { "Identifier": { "ID": "10", "Type": "External" }, "Index": "1" }

BlockStatus

Complex type representing a user's block status.

- Type: BlockStatus
- Example: { "Comment": "String", "IsBlocked": true, "Reason": "String" }

BIDefaultUser

The BI default user name for the Hyperspace user, which will be used by Hyperspace to connect to BI applications.

- Type: String
- Example: “String"

CategoryReportGrouper1

A list of values for the user's first category report grouper item, which does not have an Epic-defined meaning.

- Type: String
- Example: “String"

CategoryReportGrouper2

A list of values for the user's second category report grouper item, which does not have an Epic-defined meaning.

- Type: String
- Example: “String”

CategoryReportGrouper3

A list of values for the user's third category report grouper item, which does not have an Epic-defined meaning.

- Type: String
- Example: “String”

CategoryReportGrouper4

A list of values for the user's fourth category report grouper item, which does not have an Epic-defined meaning.

- Type: String
- Example: “String”

CategoryReportGrouper5

A list of values for the user's fifth category report grouper item, which does not have an Epic-defined meaning.

- Type: String
- Example: “String”

CategoryReportGrouper6

A list of values for the user's sixth category report grouper item, which does not have an Epic-defined meaning.

- Type: String
- Example: “String”

ContactComment

Contact comment that will appear for the initial contact.

- Type: String
- Example: “String”

ContactDate

Date to use for the newly created contact. Defaults to current date if not set.

- Type: String
- Example: “String”

CustomUserDictionaries

List of new values for custom user dictionary files.

- Type: String
- Example: “String”

Deactivated

Signifies a user should no longer have access to the application.

- Type: String
- Example: “String”

DefaultLoginDepartmentID

The login department to use for this user by default.

- Type: IDType
- Example: { "ID": "String", "Type": "String" }

DepartmentFilterSetting

Whether the department filter list is inclusive or exclusive. The default value is exclusive.

- Type: String
- Example: “String”

EmailAddress

User’s e-mail address.

- Type: String
- Example: “String”

EmployeeDemographics

Customizable items used for demographic information about a user specific to a contact.

- Type: EmployeeDemographics[]
- Example: { "EmployeeDemographic1": "String", "EmployeeDemographic2": "String", "EmployeeDemographic3": "String", “Index”: "String" }

EndDate

The date at which the user becomes inactive. Format of ‘mm-dd-yyyy’.

- Type: String
- Example: “String”

ExternalIdentifiers

List of external ID items to set for this user.

- Type: ExternalIdentifier[]
- Example: { "Identifier": "String", "IdentifierType": "String", "IsActive": true }

ExternalPasswords

Array of passwords to set for specific external ID types.

- Type: ExternalPassword[]
- Example: { "IDType": "externalIdType", "Password": "password" }

FacilityDepartmentList

A list of the departments.

- Type: IndexedRecordIn
- Example: { "Identifier": { "ID": "10", "Type": "External" }, "Index": "1" }

IdentityIDs

List of Identity IDs to assign to the user.

- Type: IDType[]
- Example: { "ID": "String", "Type": "String" }

InBasketClassifications

List of values for user In Basket classifications.

- Type: String
- Example: “String”

IsActive

Whether the user record should be set to active or inactive.

- Type: Boolean
- Example: true

LDAPOverrideID

A string that can be provided to identify the user to the LDAP server in place of the SystemLogin. Need not be unique. Maximum length 254 characters.

- Type: String
- Example: “String”

LinkedProviderID

Provider record to link to this user record.

- Type: IDType[]
- Example: { "ID": "String", "Type": "String" }

LinkedTemplatesConfig

The linked template setup for the user.

- Type: LinkedTemplatesConfig
- Example: { "AppliedTemplateID": { "ID": "String", "Type": "String" }, "DefaultTemplateID": { "ID": "String", "Type": "String" }, "AvailableLinkableTemplates": [{ "EndDate": "String", "StartDate": "String", "LoginTypes": "String", "LinkedTemplateID": { "ID": "String", "Type": "String" } }] }

LoginDepartmentFilterList

The list of departments to use when limiting access for the user.

- Type: IndexedRecordIn[]
- Example: { "Identifier": { "ID": "10", "Type": "External" }, "Index": "1" }

Name

User's name in LAST, FIRST MI format. If a UserComplexName is provided this is ignored. If one is not, this is required.

- Type: String
- Example: “String”

Notes

Up to 2,000 characters of free text notes about the user.

- Type: String
- Example: “String”

PagerID

The pager ID value for the user.

- Type: String
- Example: “String”

PhoneNumber

User’s phone number.

- Type: String
- Example: “String”

PreferredLoginDepartments

The departments on the user's preferred list.

- Type: IndexedRecordIn[]
- Example: { "Identifier": { "ID": "10", "Type": "External" }, "Index": "1" }

PrimaryManager

The primary manager of this user.

- Type: IDType
- Example: { "ID": "String", "Type": "String" }

ProviderAtLoginOption

How the user should be prompted for a provider when logging in.

- Type: String
- Example: “String”

ReceiveExternalEmail

This controls whether users receive notification emails from EpicCare link.

- Type: Boolean
- Example: true

ReceiveGroupNotifications

This controls whether users receive group notification emails from EpicCare link.

- Type: Boolean
- Example: true

ReportAuthorizedDepartmentGroups

A list of department groups for which the user has access.

- Type: IndexedRecordIn[]
- Example: { "Identifier": { "ID": "10", "Type": "External" }, "Index": "1" }

ReportAuthorizedDepartments

A list of departments for which the user has access.

- Type: IndexedRecordIn[]
- Example: { "Identifier": { "ID": "10", "Type": "External" }, "Index": "1" }

ReportAuthorizedLocations

A list of locations for which the user has access.

- Type: IndexedRecordIn[]
- Example: { "Identifier": { "ID": "10", "Type": "External" }, "Index": "1" }

ReportAuthorizedProviders

A list of providers (SER) for which the user has access.

- Type: IndexedRecordIn[]
- Example: { "Identifier": { "ID": "10", "Type": "External" }, "Index": "1" }

ReportAuthorizedServiceArea

A list of service areas for which the user has access.

- Type: IndexedRecordIn[]
- Example: { "Identifier": { "ID": "10", "Type": "External" }, "Index": "1" }

ReportAuthorizedUsers

A list of users (EMP) for which the user has access.

- Type: IndexedRecordIn[]
- Example: { "Identifier": { "ID": "10", "Type": "External" }, "Index": "1" }

ReportGrouper1

Value for the user's first free-text report grouper item, which does not have an Epic-defined meaning.

- Type: String
- Example: “String”

ReportGrouper2

Value for the user's second free-text report grouper item, which does not have an Epic-defined meaning.

- Type: String
- Example: “String”

ReportGrouper3

Value for the user's third free-text report grouper item, which does not have an Epic-defined meaning.

- Type: String
- Example: “String”

Sex

The sex of the user.

- Type: String
- Example: “Male”

SiteManagerContexts

This links users to EpicCare Link user context groups for the purposes of site management.

- Type: String[]
- Example: “String”

StartDate

The date on which the user becomes active. Defaults to the current date if not set. Format of ‘mm-dd-yyyy’.

- Type: String
- Example: “String”

SystemLoginID

Value for the user's system login. Must be unique. Maximum length 254 characters.

- Type: String
- Example: “String”

UserAlias

Value for the user's alias.

- Type: String
- Example: “String”

UserComplexName

Individual pieces of a user's name. If UserName is not provided, this is required. If this is given, it will be used instead of UserName.

- Type: PersonName
- Example: { "FirstName": "String", "GivenNameInitials": "String", "LastName": "String", MiddleName": "String" }

UserContexts

This links users to EpicCare Link user context groups.

- Type: String[]
- Example: “String”

UserDictionaryPath

Value of the path to user dictionary file. Starting in Epic May 2020, this element is no longer supported.

- Type: String
- Example: “String”

UserGroups

The values for the User Groups item.

- Type: String[]
- Example: “String”

UserInternalID

An internal ID to assign for the new user, or * to automatically generate one.

- Type: String
- Example: “String”

UserPhotoPath

A URL or file path identifying the location of a picture to show for this user. Maximum length 260 characters.

- Type: String
- Example: “String”

UserRoleIDs

List of default roles for the user. The role descriptor can be used to identify a record by selecting an ID Type of Alias.

- Type: IndexRecordIn[]
- Example: { "Identifier": { "ID": "10", "Type": "External" }, "Index": "1" }

UsersManagers

List of this user's managers. If provided, should include the PrimaryManager.

- Type: IDType[]
- Example: { "ID": "String", "Type": "String" }

UserSubtemplateIDs

The subtemplates linked to this user. Subtemplates with a lower index take priority. Maximum of 7 allowed.

- Type: IndexRecordIn[]
- Example: { "Identifier": { "ID": "10","Type": "External" }, "Index": 1" }

WebExternalIdentifier

External system Login ID.

- Type: String
- Example: “String”

E Appendix E – Supported Epic SER Attributes

The following Epic SER attributes are supported to read by Epic for query operations:

Active

Whether the practitioner's record is in active use.

- Type: Boolean
- Example: true

Communication

A language the practitioner is able to use in patient communication.

- Type: CodeableConcept[] (coding, text)
- Example: { “coding”: { “code”: “”, “display”: “”}, “text”: “” }

Gender

The gender of the practitioner.

- Type: String
- Examples: “Male”, “Female”, “Unknown”

Identifier

The practitioner or user ID.

- Type: Identifier[] (id, idtype, ini, system, type, use, value)

Name

The name of the practitioner, including all prefixes and suffixes available.

- Type: HumanName[] (family, given[], prefix, suffix, text)

Photo

Contains parameters related to the practitioner photo.

- Type: Attachment[] (url)

PractitionerRole

Contains parameters related to the practitioner photo.

- Type:PractitionerRole (active, code, location, practitioner, speciality, telecom)
- Example: { "resourceType": "PractitionerRole", "id": "DcYFyJ3YM", "active": true, "practitioner": { "reference": "Practitioner/emDkOufD", "display": "Joe Smith" }, "code": [{ "coding": [{ "code": "1", "display": "Physician" }], "text": "Physician" }], "specialty": [{ "coding": [{ "code": "19", "display": "Family Practice" }], "text": "Family Practice" }, { "coding": [{ "code": "32", "display": "Internal Medicine" }], "text": "Internal Medicine" }] }

Qualification

The practitioner's qualifications.

- Type: Qualification[] (code[])

Telecom

Returned only for Epic user-based practitioners. Not returned in a MyChart context.

- Type: ContactPoint[] (system, value)

F Appendix F – Special SER Attributes

The following Epic SER attributes require special handling for Subscriber Channel operations:

Epic Provider Address (21000 attributes)

In Epic the 21000 attributes store provider address information. This information includes telephone, fax, and email attributes. For those who are accustomed to thinking in OpenText eDirectory terms, the attributes can best be described as a multi-valued, structured attribute. There are two keys to the addressing or updating of these attributes. These are the 21000 field which is assigned by Epic, and the 21200 field which is the external identifier. It is through this external identifier that the driver tracks updates to the address record.

The primary attributes that the driver manages are:

Epic Field	Description
21000	Address Unique ID*
21010	Address Street line 1
21020	Address Street line 2
21030	Address Street line 3
21040	Address City
21050	Address State
21060	Address Zip Code
21080	Address Country
21090	Address Is Primary
21110	Address Phone
21120	Address Fax
21130	Address EMail
21200	Address External ID

**The driver passes a value of "*" (star) to Epic for 21000 on every event. This is the recommendation from Epic for creating and managing provider address records. It is intended by Epic that Address External ID (21200) is the unique identifier for managing addresses via external means (API calls).*

The driver updates/manages these fields with the Epic SER Default Configuration package. Additional fields are supported but require customization of the default package.

Global Configuration Values (GCVs)

The management of the provider address fields is configured in the SER Default Configuration GCVs. If the providers in the organization only have a single address, the GCV for “Do providers have a single location or multiple locations” (pdsepicsdcfg.address.type) should be set to “Single.” If providers could have multiple locations the GCV should be set to “Multiple.”

Single Address

If “single” is chosen, the GCVs will expand to provide a list of the standard supported attributes that are mapped through to the SER record. These are pre-populated with the default attributes in eDirectory that store this information. These attributes can be changed to fit what is appropriate for your organization.

Multiple Addresses

If “multiple” is chosen a single GCV is exposed; “Provider Address Attribute” (pdsepicsdcfg.address.multiple.attr). Provide the attribute that contains the provider address(s) in this GCV. This attribute must be a multi-valued string attribute that contains the address in JSON format. Each value must be a JSON string containing the address information for a specific facility.

The format for the JSON string is as follows:

```
{
"addressID":"IDM-1",
"isPrimary":"1",
"active":"1",
"addr1":"123 Main St",
"addr2":"Suite 1",
"addr3":"Main Hospital",
"city":"Atlanta",
"state":"GA",
"zip":"12345",
"country":"USA",
"phone":"555.123.1234",
"fax":"555.321.4321",
"email":"name@hospital.com"
}
```

Within the JSON string the “addressID” element is mapped to the “Address External ID” (21200) in Epic. This value must be unique per address for each user. But the External ID may be re-used across multiple users.

NOTE: If the organization only has a single address, but it is stored in a single attribute in eDirectory, choose “Multiple” for the address type and ensure the attribute conforms to the JSON format as specified above.

Facsimile Telephone Number

Epic will only accept a single value for each field in the address record. Facsimile Telephone Number is a multi-valued structured attribute in eDirectory. The default configuration for the driver will extract the first value from Facsimile Telephone Number and write it to the Epic 21120 field. This is only applicable when “Single” is the value chosen for the address type.