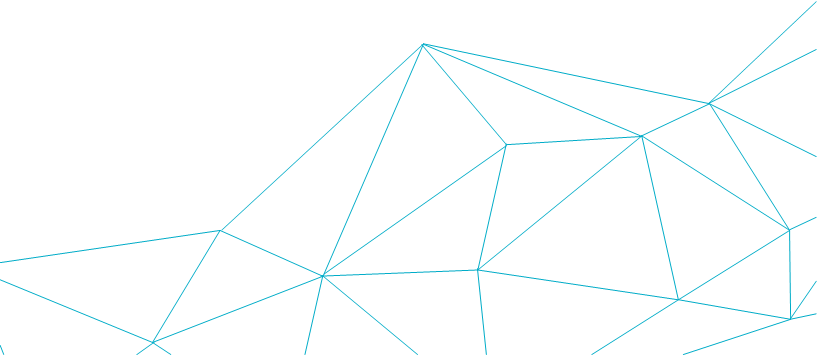
///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////



Core – Support Playbook

This document contains unpublished confidential and proprietary information of Odessa. No disclosure or use of any of these materials may be made without the expressed written consent of Odessa. ©1998-2022 Odessa. All Rights Reserved.



Document revision history.

|  |  |  |
| --- | --- | --- |
| Date | Description | Authors |
|  | Created | Pavithra Selvaraj |

­­

Contents

[1. Introduction 4](#_Toc140681060)

[1.1. Objective of this Document 4](#_Toc140681061)

[1.2. Intended Audience 4](#_Toc140681062)

[1.3. Out Of Scope: 4](#_Toc140681063)

[2. Arch Diagram: 5](#_Toc140681064)

[3. Configuration: Alerts & Notifications 5](#_Toc140681065)

[3.1. Odessa Push Logs 5](#_Toc140681066)

[3.2. Alerts 5](#_Toc140681067)

[3.3. NewRelic & ServiceNow: 6](#_Toc140681068)

[4. Understanding Errors in detail 6](#_Toc140681069)

[4.1. Error Classifications 6](#_Toc140681070)

[4.2. Inbound API Logs 6](#_Toc140681071)

[4.3. Outbound API Logs 7](#_Toc140681072)

[4.4. Incoming Message Logs 7](#_Toc140681073)

[4.5. Outgoing Message Logs 8](#_Toc140681074)

[4.6. Job Logs 9](#_Toc140681075)

[4.6.1. Core Jobs 9](#_Toc140681076)

[4.6.2. Datahub Extract Job 11](#_Toc140681077)

[4.6.3. ETL 14](#_Toc140681078)

[4.7. Background Event Monitor Logs 15](#_Toc140681079)

[4.8. Exception Logs 17](#_Toc140681080)

[5. Support Action: 17](#_Toc140681081)

# Introduction

## Objective of this Document

This playbook provides details on what are the different type of logs generated by Odessa Core Application, how they are pushed to Azure Log Analytics and how to get more details of the error.

## Intended Audience

Odessa Core Error Monitoring & Support Team.

## Out Of Scope:

The intention of this document is to cover the Application & Integration Errors.

# Arch Diagram:

A picture containing text, diagram, screenshot, plan

Description automatically generated

# Configuration: Alerts & Notifications

## Odessa Push Logs

* Odessa PushLogs is an Azure Function App responsible for moving the different types of error logs available in Core DB to Azure Log Analytics Workspace.
* Depending on the error classification, the errors are pushed into respective Custom Log tables. More details on the error classification are provided further in the document.
* Key information like Error Correlation Id, InterfaceId, PodId, Error Generated Time, Release Version Number of Core App can be found in the custom logs.
* This is currently configured to pick the errors every 2 minutes and push to Azure.

## Alerts

* Azure Alters are configured to be triggered upon any errors pushed from Core DB.
* More details on different types of alerts configured can be found in ‘Alert Rules – Core.xlsx’ (Click [here](https://hpe.sharepoint.com/:x:/r/sites/msteams_a09baa/Shared%20Documents/MOC%20-%20Management%20Of%20Change/Support/IT%20Support/Odessa%20Core/Alert%20Rules%20-%20Core.xlsx?d=w2f86750a5989454287978160a7223d78&csf=1&web=1&e=EW56SD)).
* Each alert consists of errors grouped by InterfaceId/Job Task Name/Event name depending on the error classification. *(Ref ‘Alert Dimension’ column in Alert Rules – Core.xlsx)*
* Each alert is set up to do 2 Actions.
  1. Email Notification: An Email Notification will be triggered to the configured azure user group.
  2. Function Trigger: A function app is triggered to push the errors from Azure Log Analytics Workspace to NewRelic.

## NewRelic & ServiceNow:

* Upon the error alerts triggered, corresponding error information is logged into NewRelic System.
* Further, a Service Now ticket is created based on matching conditions setup. *(For more information on the matching conditions, refer to ‘Matching Conditions’ tab in* [*Alert Rules – Core.xlsx*](https://hpe.sharepoint.com/:x:/r/sites/msteams_a09baa/Shared%20Documents/MOC%20-%20Management%20Of%20Change/Support/IT%20Support/Odessa%20Core/Alert%20Rules%20-%20Core.xlsx?d=w2f86750a5989454287978160a7223d78&csf=1&web=1&e=DexXxI)*)*

# Understanding Errors in detail

## Error Classifications

Below are the different types of Error Logs generated by Odessa Core Application:

1. Inbound API Logs
2. Outbound API Logs
3. Incoming Message Logs
4. Outgoing Message Logs
5. Job Logs
6. Background Event Monitor Logs
7. Exception Logs

Follow the steps below to get more information on the errors.

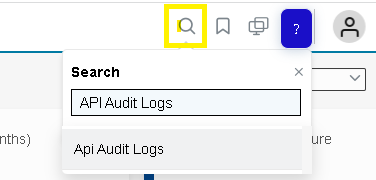
## Inbound API Logs

1. This error indicates a fault in processing the incoming API request reaching Core Application.
2. The below column values in the Azure Alert Query Result indicate that the given error belongs to this classification.

|  |  |
| --- | --- |
| Column Name | Value |
| App Name | Odessa Core |
| Component Type | Inbound API |

1. To get more information on the error, the below steps are to be followed:

* Login into Odessa Core Application
* Go to ‘API Audit Logs’ using the Search Menu option.



* Set Filter value as below and hit search:
  + Select Parameter: Correlation Id
  + Enter Value: TransactionId from the Azure Alert Query Result
* To get more details on the error, select the row from the result and click on ‘View’ button available on top left corner of result grid.

A close-up of a computer screen

Description automatically generated with low confidence

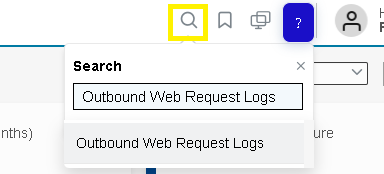
## Outbound API Logs

1. This error indicates a fault in outgoing API requests originating from Core Application.
2. The below column values in the Azure Alert Query Result indicate that the given error belongs to this classification.

|  |  |
| --- | --- |
| Column Name | Value |
| App Name | Odessa Core |
| Component Type | Outbound API |

1. To get more information on the error, the below steps are to be followed:

* Login into Odessa Core Application
* Go to ‘Outbound Web Request Logs’ using the Search Menu option.



* Set Filter value as below and hit search:
  + Select Parameter: Correlation Id
  + Enter Value: TransactionId from the Azure Alert Query Result
* A screenshot of a computer

  Description automatically generated with medium confidenceTo get more details on the error, select the row from the result and click on ‘View’ button available on top left corner of result grid.

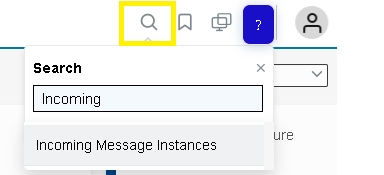
## Incoming Message Logs

1. This error indicates a fault in processing a Message Received from the Message Broker End Points configured (Azure Service Bus: Topic/Queue).
2. The below column values in the Azure Alert Query Result indicate that the given error belongs to this classification.

|  |  |
| --- | --- |
| Column Name | Value |
| App Name | Odessa Core |
| Component Type | Inbound Message |

1. To get more information on the error, the below steps are to be followed:

* Login into Odessa Core Application
* Go to ‘Incoming Message Instances’ using the Search Menu option.



* Set Filter value as below and hit search:
  + Select Parameter: Correlation Id
  + Enter Value: TransactionId from the Azure Alert Query Result
* A screenshot of a computer

  Description automatically generated with low confidenceTo get more details on the error, select the row from the result and click on ‘View’ button available on top left corner of result grid.

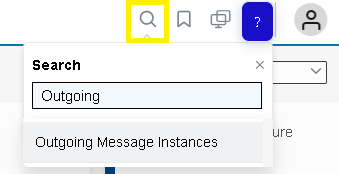
## Outgoing Message Logs

1. This error indicates a fault while sending a Message to the Message Broker End Points configured (Azure Service Bus: Topic/Queue).
2. The below column values in the Azure Alert Query Result indicate that the given error belongs to this classification.

|  |  |
| --- | --- |
| Column Name | Value |
| App Name | Odessa Core |
| Component Type | Outbound Message |

1. To get more information on the error, the below steps are to be followed:

* Login into Odessa Core Application
* Go to ‘Outgoing Message Instances’ using the Search Menu option.



* Set Filter value as below and hit search:
  + Select Parameter: Correlation Id
  + Enter Value: TransactionId from the Azure Alert Query Result
* A screenshot of a computer

  Description automatically generated with medium confidenceTo get more details on the error, select the row from the result and click on ‘View’ button available on top left corner of result grid.

## Job Logs

## Core Jobs

1. This error indicates a fault while running a Job. The error is logged at Job Step Instance Level (i.e., Task Level)
2. The below column values in the Azure Alert Query Result indicate that the given error belongs to this classification.

|  |  |
| --- | --- |
| Column Name | Value |
| App Name | Odessa Core |
| Component Type | Job Step Instance |

*\*This section is applicable for all jobs except DataHub/Insight related, which is discussed further in the same document.*

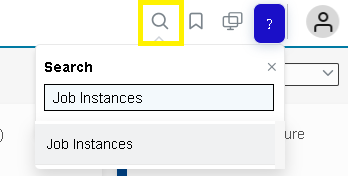
*\*Operation Name = “DataHub Extract” is the Datahub job*

*\* Operation Name = “ETL” is the Insight job*

*\*Any other Operation Name is a Core* *Job and the following instructions are applicable*

1. To get more information on the error, the below steps are to be followed:

* Login into Odessa Core Application
* Go to ‘Job Instances’ using the Search Menu option.

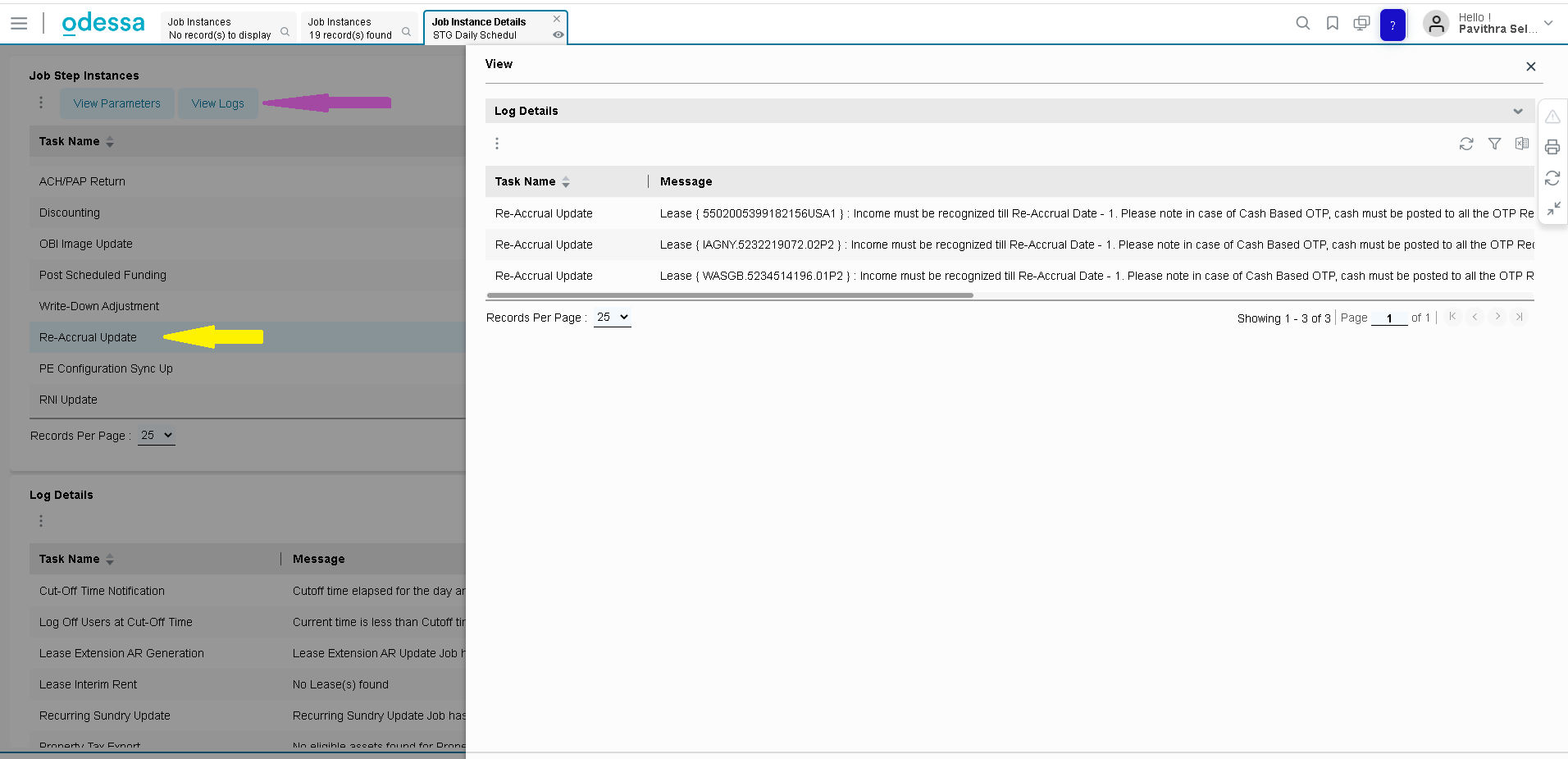


* Remove any preset filter that are defaulted and Click on Advance Search
* Enter the Filter value as below and hit search:
  + Filter By: Job Instance Id
  + Condition: Equals
  + Enter Value: TransactionId from the Azure Alert Query Result

A screenshot of a computer

Description automatically generated with medium confidence

* To get more details on the error,
  + Select the row from the result and click on ‘View Job Instance Details’ button available on top left corner of result grid.
  + In the Job Instance Details screen, in the Job Step Instances grid look for Task Name with Operation Name found in Azure Result Query. *(ref. Yellow arrow in below screenshot)*
  + Select the row and click on View Logs *(ref. Purple arrow in below screenshot)*



## Datahub Extract Job

1. This error indicates a fault while running a Job. The error is logged at Job Step Instance Level (i.e., Task Level)
2. The below column values in the Azure Alert Query Result indicate that the given error belongs to this classification.

|  |  |
| --- | --- |
| Column Name | Value |
| App Name | Odessa Core |
| Component Type | Job Step Instance |
| Operation Name | Datahub Extract |

3. To get more information on the error, the below steps are to be followed:

* Login into Odessa Core Application
* Go to ‘Job Instances’ using the Search Menu option.

A screenshot of a computer

Description automatically generated with medium confidence

* Remove any preset filter that are defaulted and Click on Advance Search
* Enter the Filter value as below and hit search:
  + Filter By: Job Instance Id
  + Condition: Equals
  + Enter Value: TransactionId from the Azure Alert Query Result

A screenshot of a computer

Description automatically generated with medium confidence

1. Error message can be found here in case of below failure cases,

* Core web job failed to communicate to Primary/ Secondary ODW VM (or) Existing Job not finished (or) Agent Job is not running.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

1. In case of failures due to SQL connectivity or other SQL related issues like Timeout,.. error message can be found in the Datahub Engine deployed path in ODW VM like below,

A screenshot of a computer

Description automatically generated

\*\*RDC to ODW server to find the error files.

A screenshot of a computer

Description automatically generated with medium confidence

1. In case of failures at an entity level like Primary key violation or metadata error, message can be found in ODH database using below Query,

DECLARE @EtlType NVARCHAR(20) = 'Partial', @ETLJobInstanceId INT

SET @ETLJobInstanceId = (SELECT Top 1 Id from OETLCONFIG.ETLJobInstances WHERE Status = 'Failed' AND ETLRunType = @EtlType ORDER BY 1 DESC)

SELECT ErrorMessage, \* FROM OETLCONFIG.ETLJobEntityInstances WHERE ETLJobInstanceId = @ETLJobInstanceId

A screenshot of a computer

Description automatically generated with low confidence

## ETL

1. This error indicates a fault while running a Job. The error is logged at Job Step Instance Level (i.e., Task Level)
2. The below column values in the Azure Alert Query Result indicate that the given error belongs to this classification.

|  |  |
| --- | --- |
| Column Name | Value |
| App Name | Odessa Core |
| Component Type | Job Step Instance |
| Operation Name | ETL |

3. To get more information on the error, the below steps are to be followed:

* Login into Odessa Core Application
* Go to ‘Job Instances’ using the Search Menu option.

A screenshot of a computer

Description automatically generated with medium confidence

* Remove any preset filter that are defaulted and Click on Advance Search
* Enter the Filter value as below and hit search:
  + Filter By: Job Instance Id
  + Condition: Equals
  + Enter Value: TransactionId from the Azure Alert Query Result

A screenshot of a computer

Description automatically generated with medium confidence

1. Error message can be found here in case of below failure cases:
   1. Core web job failed to communicate to Data Warehouse VM/Database Server.

If these situations occur, the ETL task will be labeled as "Faulted," and the Exception message will display "Connection to the server could not be established". This will require coordinating with HPE IT team to resolve the connectivity issue.A screenshot of a computer

Description automatically generated

* 1. Failed while running the dimension processing – when loading OLTP tables into dimension tables in Data Warehouse.

A screenshot of a computer

Description automatically generated

To obtain additional information when encountering an error in the ETL steps, you can access the Data Warehouse Database Server and execute the provided query. This query will provide further details regarding the specific Entity that failed to undergo processing.

-- Replace Execution Id with what you are seeing Odessa Core Job logs.

select \* from EtlExecutionJobLogs where ExecutionId=<Execution Id> AND Level IN ('Fatal', 'Error')

A screenshot of a computer

Description automatically generated

* 1. Failed while running the facts processing.

A screenshot of a computer

Description automatically generated

To obtain additional information when encountering an error in the ETL steps, you can access the Data Warehouse Database Server and execute the provided query. This query will provide further details regarding the specific Entity that failed to undergo processing.

-- Replace Execution Id with what you are seeing Odessa Core Job logs.

select \* from EtlExecutionJobLogs where ExecutionId=<Execution Id> AND Level IN ('Fatal', 'Error')

A screenshot of a computer

Description automatically generated

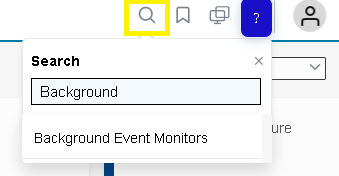
## Background Event Monitor Logs

1. This error indicates a fault while processing a Background Event.
2. The below column values in the Azure Alert Query Result indicate that the given error belongs to this classification.

|  |  |
| --- | --- |
| Column Name | Value |
| App Name | Odessa Core |
| Component Type | Background Event |

1. To get more information on the error, the below steps are to be followed:

* Login into Odessa Core Application
* Go to ‘Background Event Monitors’ using the Search Menu option.



* Click of Advanced Search and filter as below and click Search:
  + Select Parameter: Correlation Id
  + Filter Type: Equals
  + A screenshot of a computer

    Description automatically generated with medium confidenceValue: TransactionId from the Azure Alert Query Result
* To get more details on the error, select the row from the result and click on ‘View’ button available on top left corner of result grid.

A screenshot of a computer

Description automatically generated with medium confidence

Note: Every faulted background event will be rerun/reprocessed once again at the end of day automatically by the system. However, this can also be triggered adhoc anytime by running the “Background Monitor” job (This job is always a system level job that will rerun all the faulted background processes).

## Exception Logs

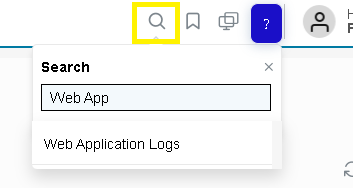
1. This error indicates a fault or crash observed while performing any Core application operations.
2. The below column values in the Azure Alert Query Result indicate that the given error belongs to this classification.

|  |  |
| --- | --- |
| Column Name | Value |
| App Name | Odessa Core |
| Component Type | App Exceptions |

Sample: <To-Do>

1. To get more information on the error, the below steps are to be followed:

* Login into Odessa Core Application
* Go to ‘Web Application Logs’ using the Search Menu option.



* Set Filter value as below and hit search:
  + Select Parameter: Correlation Id
  + Enter Value: TransactionId from the Azure Alert Query Result
* To get more details on the error, select the row from the result and click on ‘View’ button available on top left corner of result grid.

A screen shot of a computer

Description automatically generated with low confidence

# Support Action:

* Transaction support team will analyze the issue with the information available based on the steps mentioned above and take it forward as per the MVP “Support Model” Plan.
* Odessa team will help them analyze the errors on the Hyper Care period for the support team to get Hands-On.
* “Support Model” plan will cover details on
  1. Steps to create JIRA for Odessa
  2. Model to provide/support data needed for debugging to Odessa
  3. Defect Lifecycle & Closure