

Informatica 10.5.1 Logging Guide

Contents

1. Purpose of Document	2
1.1 Security Parameters Required	2
2. Admin Console Configuration Steps.....	3
2.1 Repository Service Properties	3
2.2 Integration Service Properties.....	5
3. Workflow Manager Configuration	6
3.1 Workflow Properties.....	6
3.2 Session Logs Configuration.....	9
3.3 Default Session Logs Directory.....	11
3.4 Default Workflow Logs Directory.....	12
3.5 Normal vs Verbose Data Loggings	13
4. Getting other logs via Command Line	16
4.1 Fetch All Domain Logs.....	16
4.2 Fetch Repository Service Logs (REPO_ORACLE).....	16
4.3 Fetch Integration Service Logs (INT_SERV_ORACLE)	17
4.4 Fetch All Logs (Full System)	17
4.5 Fetch Only Error and Critical Logs.....	17
4.6 Fetch Logs Within a Specific Timeframe (Last 30 Days).....	17
4.7 Batch File to Run All Commands at Once.....	18
4.8 Summary of Output Log Files	19

1. Purpose of Document

This document serves as a comprehensive guide for configuring, monitoring, and retrieving logs in **Informatica 10.5.1**. It outlines the **security parameters, admin console configurations, and workflow/session logging mechanisms** necessary to maintain system integrity, ensure compliance, and support troubleshooting.

Key objectives of this document:

- Define **security logging requirements** to capture relevant user activities, system events, and potential security breaches.
- Explain **Admin Console configurations** for Repository Service and Integration Service to optimize logging and retention policies.
- Detail **workflow and session log configurations**, including normal vs. verbose logging, log file naming conventions, and storage management.
- Provide **command-line methods** to fetch domain-wide, repository, integration service, workflow, and session logs for auditing or debugging.
- Facilitate **log management best practices** to balance **detailed auditing with storage efficiency**.

1.1 Security Parameters Required

Please see the prerequisite requirement below, kindly share the sample of **access and audit trail logs** to initial validation, please ensure that the logs to be ingested into Splunk contain the following security parameters.

Source IP – Source IP of the user from where he is logged-in.

Login Action (Failure/Success) – Login action success and in case failure we need to record the reason (like bad password) and the user ID/name as well.

Password changes – in case user has requested for password change this must be recorded as well.

User/Admin activities – actual activity performed by the user.

Destinations – the destination application name/url/host to which the user has requested to access.

Application changes – same as activity

Addition/modification/deletion at the record and field level – same as activity.

Date & Time – Actual date & time of the user login/activity, note: all the events must have data/time.

Username – Along with IP of the user we need username/userID to be recorded.

Time Sync with NTP – Application must be synced with NTP server to record correct time.

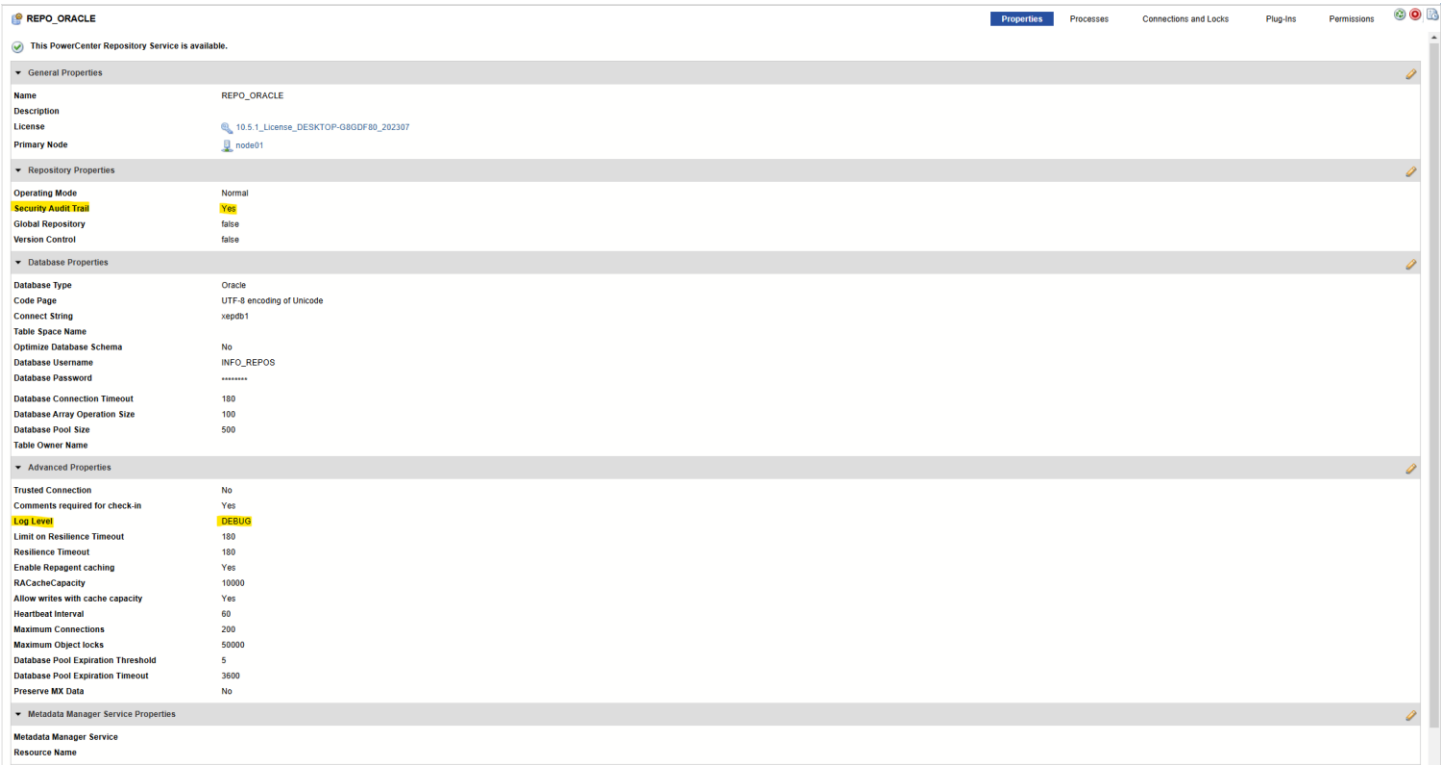
The snippet from an email outlines the prerequisite security parameters required for **access and audit trail logs** to be ingested into **Splunk** for validation. It ensures that the logs capture key security details such as **Source IP, Login Actions (Success/Failure), Password Changes, User Activities, Destinations, Application Modifications, and Timestamps**. The inclusion of **NTP**

time synchronization ensures logs maintain accuracy and integrity. This request was sent to verify that the logging mechanism meets **security and compliance standards**, ensuring all critical activities are **tracked, auditable, and useful for forensic analysis**.

2. Admin Console Configuration Steps

These configurations should be done on the Admin Console (webpage):

2.1 Repository Service Properties



Explanation of Highlighted Parameters in Repository Service (Admin Console):

1. Security Audit Trail (Yes)

- **Description:** When enabled, this setting ensures that all critical user actions and system changes are logged for auditing purposes.
- **Security Benefits:**
 - **Tracks user/admin activities** such as logins, configuration changes, and repository modifications, aligning with "**User/Admin activities**".
 - Records sensitive actions like **password changes, application changes, and record-level modifications** ("**Addition/modification/deletion at the record and field level**").
 - Helps with **compliance requirements** and facilitates **incident investigations** by maintaining a detailed history of activities.

- Assists in monitoring **"Login Action (Failure/Success)"** by logging both successful and failed login attempts.

2. Log Level (DEBUG) for Repository Service

- **Description:**

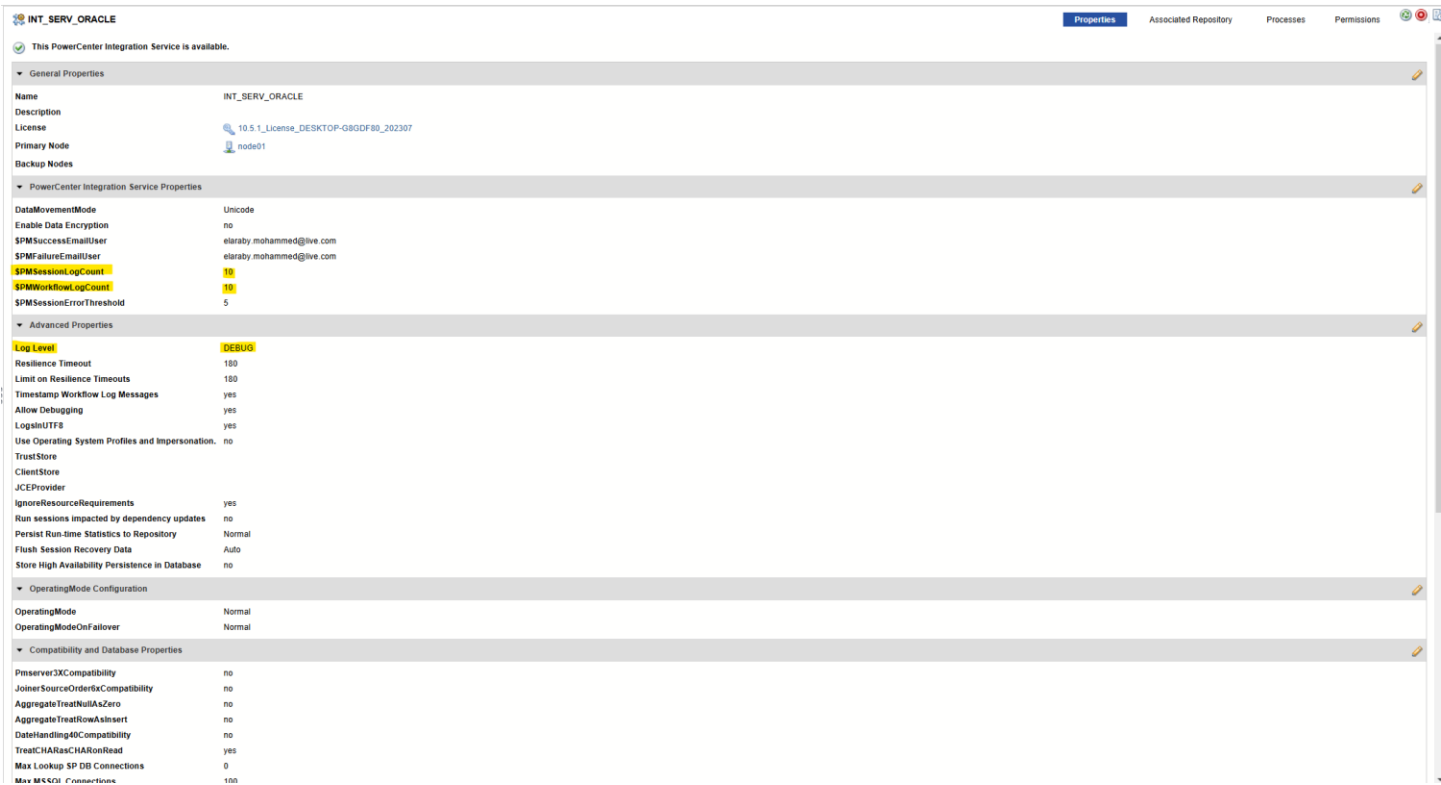
The DEBUG log level captures detailed repository events, including metadata changes, user authentication, object versioning, and configuration modifications. This provides a comprehensive view of repository interactions, aiding in troubleshooting and security monitoring.

- **Security Benefits:**

- **Tracks all user actions within the repository**, ensuring visibility into **who modified objects, when, and how**, supporting **"User/Admin activities"** and **"Application changes."**
- **Captures authentication attempts**, including **successful and failed logins**, helping enforce security parameters like **"Login Action (Failure/Success)"** and **"Password changes."**
- **Logs repository object modifications**, including **record-level changes**, aligning with **"Addition/modification/deletion at the record and field level."**
- **Stores timestamps for all critical events**, supporting **"Date & Time"** logging for compliance and auditing.
- **Includes detailed user information (username, security domain, and IP address)**, enhancing security visibility for **"Source IP"** and **"Username."**
- Assists in **detecting unauthorized access attempts** and **potential security breaches** by providing a detailed audit trail.

Combined, these settings maximize visibility into user actions and system events, strengthening overall security compliance and aiding in precise incident tracking.

2.2 Integration Service Properties



Explanation of Highlighted Parameters in Integration Service (Admin Console):

1. \$PMSessionLogCount:

- **Purpose:** Sets the maximum number of **session logs** to retain.
- **Function:** When the count exceeds the limit, the oldest session logs are automatically deleted.
- **Benefit:** Maintains log history for audits while managing disk space.

2. \$PMWorkflowLogCount:

- **Purpose:** Defines the number of **workflow logs** to retain.
- **Function:** Older workflow logs are removed once the count exceeds the set limit.
- **Benefit:** Ensures log retention for troubleshooting and compliance without overusing storage.

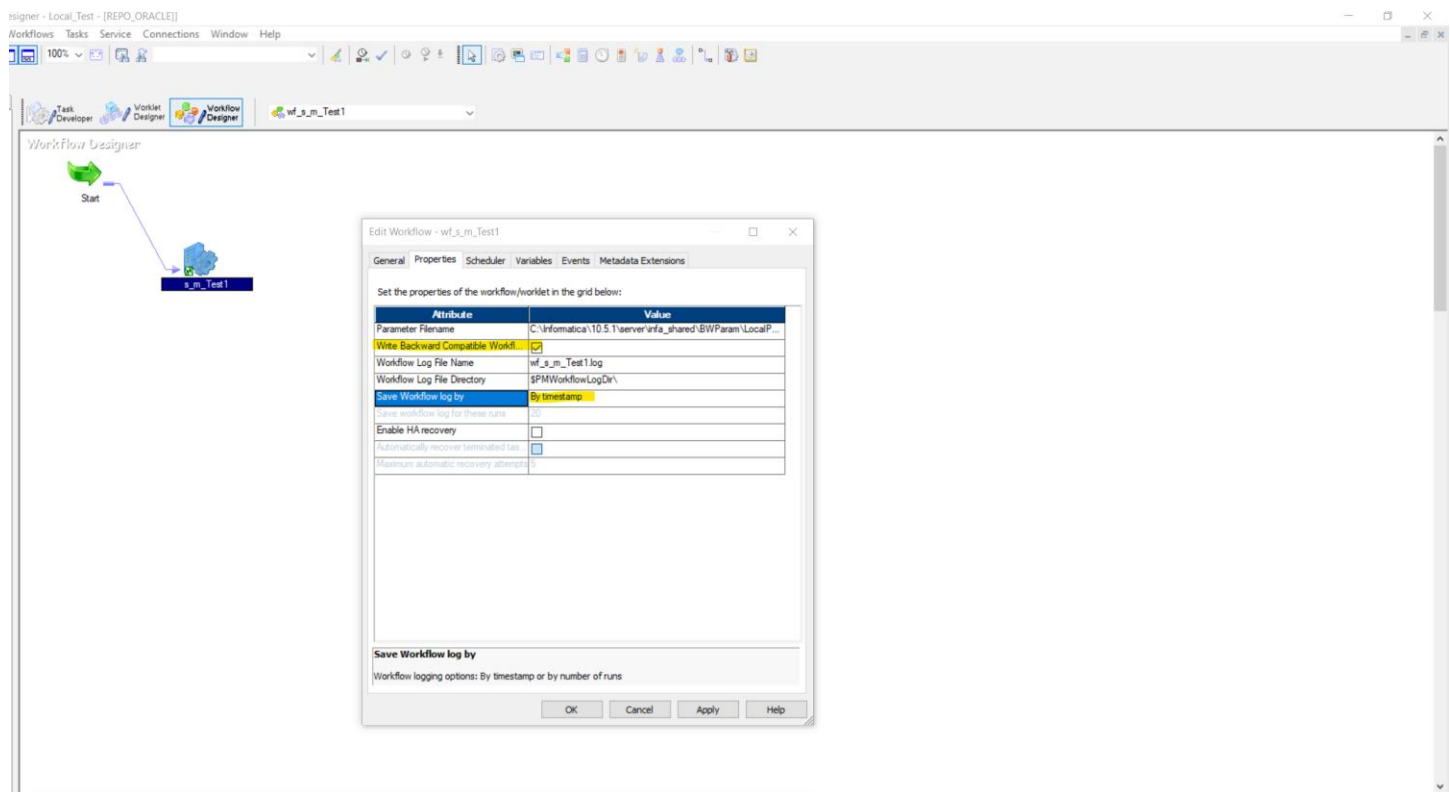
3. Log Level (DEBUG) for Integration Service

- **Description:**
The DEBUG log level captures highly detailed information about session executions, workflow events, transformations, performance metrics, and errors. This level logs every step of data movement and process execution.
- **Security Benefits:**

- Provides in-depth tracking of workflow and session execution details, aiding in detecting unauthorized modifications or failures.
- Logs detailed user activities and interactions with workflows, supporting security parameters like **"User/Admin activities"** and **"Application changes."**
- Helps identify **failed login attempts, password changes, or unauthorized data modifications**, supporting **"Login Action (Failure/Success)"** and **"Password changes."**
- Captures **source IPs, timestamps, and session details**, aligning with **"Source IP," "Date & Time,"** and **"Username"** security parameters.
- Aids in forensic analysis during security incidents by providing a complete execution trace.

3. Workflow Manager Configuration

3.1 Workflow Properties



The **"Write Backward Compatible Workflow Log File"** option in Informatica is designed to create an additional **plain text log file** that is compatible with older versions of Informatica tools and third-party log readers.

Key Functions of **"By Timestamp"**:

1. Unique Log Files for Each Run:

- The timestamp (format: YYYYMMDDHHMMSS) is added to the log file name.
- Example:

```
wf_s_m_Test1.log.20240225132801  
wf_s_m_Test1.log.20240225140022  
wf_s_m_Test1.log.20240225143245
```

2. No Overwriting:

- Each execution creates a new log file.
- Older logs are **never overwritten**, ensuring a complete history of workflow executions.

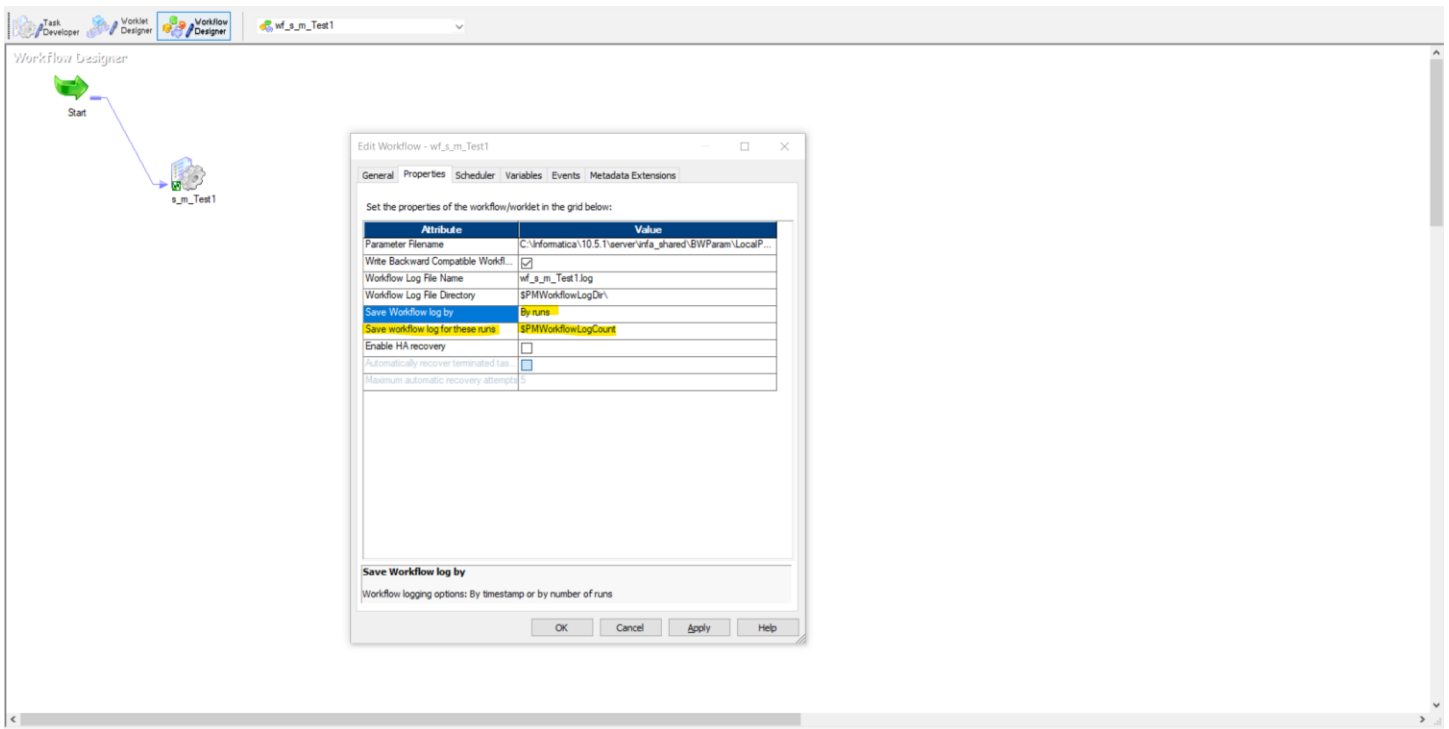
3. Unlimited Log Retention (Unless Managed):

- There is **no internal limit** on the number of logs saved when using **"By Timestamp"**.
- Retention is managed by:
 - **Manual deletion**
 - **External scripts (log rotation)**
 - **Storage capacity limits**

4. Accurate Run Identification:

- The timestamp helps identify when each workflow run was executed.
- Useful for **audits, troubleshooting, and performance analysis**.

Workflow Logs Saved by Runs



In Informatica Workflow Designer, the "Save Workflow log by" option allows you to control how workflow logs are saved:

- **By runs:** Saves logs based on the number of workflow executions.
- **By timestamp:** (alternative option) Saves logs based on the time of execution.

Highlighted Setting:

- **Save workflow log for these runs** (\$PMWorkflowLogCount):
 - This sets the number of past workflow runs for which logs are retained.
 - If set to **10** (as configured in the **Admin Console** for the Integration Service), **only the logs from the last 10 workflow executions** are kept. Older logs are automatically deleted once the limit is exceeded.

Example:

- If a workflow is run 15 times, only logs from the most recent 10 runs will be available, and logs from the first 5 runs will be purged.

This approach helps **manage disk space** while retaining logs for recent executions for review or troubleshooting.

Example of 5 Runs (Saved by Run):

Assuming the workflow is named **wf_s_m_Test1**, the log files after 5 runs would look like:

1. wf_s_m_Test1.log.1 → (Log for Run 1)
2. wf_s_m_Test1.log.2 → (Log for Run 2)
3. wf_s_m_Test1.log.3 → (Log for Run 3)
4. wf_s_m_Test1.log.4 → (Log for Run 4)
5. wf_s_m_Test1.log.5 → (Log for Run 5)

3.2 Session Logs Configuration

Workflow Designer

Task Developer | Workflow Designer | Workflow Designer

wf_s_m_Test1

Start

s_m_Test1

Edit Tasks

General | Properties | Config Object | Mapping | Components | Metadata Extensions

Select task: s_m_Test1

Task type: Session (Reusable)

Config Name: default_session_config

Attribute	Value	UnOverride
Advanced		
Constraint based load ordering	<input type="checkbox"/>	
Cache LOOKUP function	<input checked="" type="checkbox"/>	
Default buffer block size	Auto	
Line Sequential buffer length	1024	
Maximum Memory Allowed For Auto Memory Attributes	512MB	
Maximum Percentage of Total Memory Allowed For A...	5	
Additional Concurrent Pipelines for Lookup Cache Cre...	Auto	
Custom Properties		
Pre-build lookup cache	Auto	
Date/Time Format String	MM/DD/YYYY HH:MM:SS.US	
Pre 85 Timestamp Compatibility	<input type="checkbox"/>	
Log Options		
Save session log by	Session timestamp	Revert
Save session log for these runs	1	Revert
Session Log File Max Size	0	
Session Log File Max Time Period	0	
Maximum Partial Session Log Files	1	
Writer Commit Statistics Log Frequency	1	
Writer Commit Statistics Log Interval	0	
Error handling		
Stop on errors	0	
Override tracing	Normal	Revert
On Stored Procedure error	Stop	
On Pre-session command task error	Stop	
On Pre-Post SQL error	Stop	
Error Log Type	None	
Error Log DB Connection		
Error Log Table Name Prefix		
Error Log File Directory	SPM\src\FireDir\	

Save session log by

Save session log by timestamp or by number of runs.

OK Cancel Apply Help

Workflow Designer

Task Developer | Workflow Designer | Workflow Designer

wf_s_m_Test1

Start

s_m_Test1

Edit Tasks

General | Properties | Config Object | Mapping | Components | Metadata Extensions

Select task: s_m_Test1

Task type: Session (Reusable)

Config Name: default_session_config

Attribute	Value	UnOverride
Advanced		
Constraint based load ordering	<input type="checkbox"/>	
Cache LOOKUP function	<input checked="" type="checkbox"/>	
Default buffer block size	Auto	
Line Sequential buffer length	1024	
Maximum Memory Allowed For Auto Memory Attributes	512MB	
Maximum Percentage of Total Memory Allowed For A...	5	
Additional Concurrent Pipelines for Lookup Cache Cre...	Auto	
Custom Properties		
Pre-build lookup cache	Auto	
Date/Time Format String	MM/DD/YYYY HH:MM:SS.US	
Pre 85 Timestamp Compatibility	<input type="checkbox"/>	
Log Options		
Save session log by	Session timestamp	Revert
Save session log for these runs	1	Revert
Session Log File Max Size	0	
Session Log File Max Time Period	0	
Maximum Partial Session Log Files	1	
Writer Commit Statistics Log Frequency	1	
Writer Commit Statistics Log Interval	0	
Error handling		
Stop on errors	0	
Override tracing	Verbose Data	Revert
On Stored Procedure error	Stop	
On Pre-session command task error	Stop	
On Pre-Post SQL error	Stop	
Error Log Type	None	
Error Log DB Connection		
Error Log Table Name Prefix		
Error Log File Directory	SPM\src\FireDir\	

Override tracing

Tracing level for the session

OK Cancel Apply Help

In Informatica, **tracing levels** determine the amount of detail logged during session execution. Here's a brief overview of the key tracing levels:

1. Normal:

- **Description:** Logs standard initialization and status information, errors, and skipped rows due to transformation errors. Provides a summary without detailing individual rows.
- **Use Case:** Routine session monitoring with minimal performance impact.

2. Verbose Initialization:

- **Description:** Includes all details from the Normal level, plus additional initialization information, such as names of index and data files used, and detailed transformation statistics.
- **Use Case:** Useful for diagnosing initialization issues or configuration problems.






3. Verbose Data:

- **Description:** Encompasses Verbose Initialization details and logs each row processed by the session. Also notes where string data is truncated to fit column precision.
- **Use Case:** Employed for in-depth debugging to trace data flow through transformations.

Considerations:

- **Performance Impact:** Higher tracing levels, especially Verbose Data, can significantly affect performance and generate large log files.
- **Best Practices:** Use Verbose levels sparingly and revert to Normal after troubleshooting to maintain optimal performance.

3.3 Default Session Logs Directory

 > This PC > WinToUSB (C:) > Informatica > 10.5.1 > server > infa_shared > SessLogs				Search SessLogs
Name	Date modified	Type	Size	
 s_m_Test1.log.20250225151338	2/25/2025 3:13 PM	20250225151338 F...	8 KB	
 s_m_Test1.log.20250225151338.bin	2/25/2025 3:13 PM	BIN File	17 KB	
 s_m_Test1.log.20250225151039	2/25/2025 3:10 PM	20250225151039 F...	28 KB	
 s_m_Test1.log.20250225151039.bin	2/25/2025 3:10 PM	BIN File	48 KB	

1. Normal Tracing (Upper Two Files):

- **s_m_Test1.log.20250225151338 (8 KB)** — **Text log** file in **Normal** tracing mode.
 - **Timestamp:** 20250225151338 → **Feb 25, 2025, at 15:13:38**
- **s_m_Test1.log.20250225151338.bin (17 KB)** — **Binary** version of the Normal trace log.
 - **Timestamp:** Same as above.

Key Points:

- Captures **basic session events** like start, end, and general errors.
- **Smaller file size** indicates fewer logged details.
- Suitable for routine monitoring without performance impact.

2. Verbose Data Tracing (Lower Two Files):

- **s_m_Test1.log.20250225151039 (28 KB)** — **Text log** file in **Verbose Data** tracing mode.
 - **Timestamp:** 20250225151039 → **Feb 25, 2025, at 15:10:39**
- **s_m_Test1.log.20250225151039.bin (48 KB)** — **Binary** version containing detailed data points.
 - **Timestamp:** Same as above.

Key Points:

- Provides **row-level data**, transformation logic, and real-time process details.
- **Larger file size** reflects the depth of logging.
- Designed for **in-depth troubleshooting** but could impact performance if overused.

3.4 Default Workflow Logs Directory

WinToUSB (C:) > Informatica > 10.5.1 > server > infa_shared > WorkflowLogs				Search WorkflowLogs
Name	Date modified	Type	Size	
wf_s_m_Test1.log.20250225150933	2/25/2025 3:09 PM	20250225150933 F...	10 KB	
wf_s_m_Test1.log.20250225150933.bin	2/25/2025 3:09 PM	BIN File	17 KB	
wf_s_m_Test1.log.20250225151024	2/25/2025 3:10 PM	20250225151024 F...	10 KB	
wf_s_m_Test1.log.20250225151024.bin	2/25/2025 3:10 PM	BIN File	17 KB	

In the **WorkflowLogs** directory, you can see multiple workflow log files, both in **text** (.log) and **binary** (.bin) formats. Here's a breakdown of what appears:

1. Log Files (.log):

- **wf_s_m_Test1.log.20250225150933** and **wf_s_m_Test1.log.20250225151024**
 - These are plain-text log files for the workflow named **wf_s_m_Test1**.
 - Each log file is timestamped (20250225150933 and 20250225151024), indicating the date and time of execution.
 - **Size:** ~10 KB, suggesting a standard logging level (likely normal or less verbose).

2. Binary Log Files (.bin):

- **wf_s_m_Test1.log.20250225150933.bin** and **wf_s_m_Test1.log.20250225151024.bin**
 - These are the **binary versions** of the corresponding text logs.
 - **Purpose:** Used for replaying sessions in Informatica tools or for detailed analysis using Informatica’s log viewers.
 - **Size:** ~17 KB, generally larger than text logs as they contain more structured data.

3. Naming Convention:

- **wf_s_m_Test1** → Workflow Name
- **20250225150933** → Timestamp (YYYYMMDDHHMMSS)
- **.log** → Text Log
- **.log.bin** → Binary Log

3.5 Normal vs Verbose Data Loggings

Aspect	Normal Session Log	Verbose Session Log
Start Time	Tue Feb 25, 2025, 15:13:43	Tue Feb 25, 2025, 15:10:43
End Time	Tue Feb 25, 2025, 15:13:44	Tue Feb 25, 2025, 15:10:44
Log Size	Smaller (less detailed)	Larger (highly detailed)
Tracing Level	Standard session flow & key events	Full data lineage, row-level data, & expressions
Error Reporting	Summarizes errors & warnings	Detailed error tracing per row/column
Transformations Insight	Basic transformation info	Detailed row transformations, data types, defaults
Row-Level Data	Not Included	Included (Full data records)
SQL Queries & DB Actions	Summarized SQL operations	Detailed DB connection, SQL execution steps
Performance Metrics	Simple load summary	Detailed transformation row statistics
Use Case	Regular monitoring & troubleshooting	In-depth debugging, data validation, root-cause analysis

Key Differences:

- **Normal Log** focuses on session flow with essential events—**ideal for general monitoring**.
- **Verbose Log** includes **row-level tracking**, transformation details, and SQL operations—**best for deep debugging**.
- **Verbose** shows individual data records passing through transformations, while **Normal** only summarizes results.

When to Use:

- **Normal Log:** For routine execution monitoring.
- **Verbose Log:** When troubleshooting data issues, debugging complex transformations, or validating data flows.

Parameters Monitored by Session Logs (Normal vs. Verbose Logging):

Security Parameter	Normal Logging	Verbose Data Logging	Example

Source IP	✗	✓	Verbose: READER_1_1_1> Source is [localhost:1521/xepdb1], User [TEST_DB]
Login Action (Failure/Success)	✓	✓	Normal: DIRECTOR> TM_6014 Initializing session [s_m_Test1]
Password Changes	✗	✗	Not captured in both logs
User/Admin Activities	✓	✓	Normal: MANAGER> PETL_24005 Starting post-session tasks.
Destinations	✓	✓	Normal: WRITER_1_*_1> Target is Database [localhost:1521/xepdb1], User [TEST_DB]
Application Changes	✗	✓	Verbose: MAPPING> DBG_21249 Initializing Transform: EXP_ADD_DATE
Addition/Modification/Deletion (Record/Field)	✗	✓	Verbose: WRITER_1_*_1> SQL INSERT INTO TEST_TARGET(ID, NAME, VALUE, REFERENCE_DATE) VALUES (?, ?, ?, ?)
Date & Time	✓	✓	Normal: Load Start Time: Tue Feb 25 15:13:43 2025
Username	✗	✓	Verbose: READER_1_1_1> Source is [localhost:1521/xepdb1], User [TEST_DB]
Time Sync with NTP	✗	✗	Not captured in both logs

Key Insights:

- **Normal Logs** cover basic session info: success/failure, activities, destinations, and timestamps.
- **Verbose Logs** provide granular tracking: source IPs, usernames, data changes, and detailed transformations.
- **Verbose Logs** are essential for security compliance (e.g., for Splunk ingestion) due to richer metadata.

Parameters Monitored by Workflow Logs:

Security Parameter	Monitored?	Example from Log
Source IP	✗	Not captured in the provided logs.
Login Action (Failure/Success)	✓	INFO : LM_36488 ... Connected to repository [REPO_ORACLE] in domain [Domain] as user [Administrator]
Password Changes	✗	No evidence of password change tracking.
User/Admin Activities	✓	INFO : LM_36488 ... Fetching initialization properties from the Integration Service
Destinations (application/url/host)	✓	INFO : LM_36488 ... Connected to repository [REPO_ORACLE] in domain [Domain]
Application Changes	✗	Application changes like configuration updates are not tracked here.
Addition/Modification/Deletion at Record Level	✗	Data-level changes are not captured in workflow logs; session logs handle this.
Date & Time	✓	INFO : LM_36435 [Tue Feb 25 15:13:24 2025] : Starting execution of workflow [wf_s_m_Test1]
Username	✓	INFO : LM_36488 ... as user [Administrator]
Time Sync with NTP	✗	No reference to NTP time synchronization.

Examples Proving Each Monitored Parameter:

1. Login Action (Success):

- INFO : LM_36488 ... Connected to repository [REPO_ORACLE] ... as user [Administrator]
- Indicates a successful login to the repository.

2. User/Admin Activities:

- INFO : LM_36488 ... Fetching initialization properties from the Integration Service
- Shows system-level activities performed during the workflow execution.

3. Destinations (application/url/host):

- INFO : LM_36488 ... Connected to repository [REPO_ORACLE]
- Logs the destination repository accessed during the workflow.

4. Date & Time:

- INFO : LM_36435 [Tue Feb 25 15:13:24 2025] : Starting execution of workflow [wf_s_m_Test1]
- Precise timestamp for workflow start.

5. Username:

- INFO : LM_36488 ... as user [Administrator]
- Captures the username involved in the session.

Summary:

- **Monitored:** Login actions, user activities, destination repository, timestamps, and username.
- **Not Monitored:** Source IP, password changes, application changes, record-level modifications, and NTP sync.

4. Getting other logs via Command Line

This document provides a structured approach to fetching all other logs from an Informatica 10.5.1 setup. It outlines the commands required to capture logs for Domain, Repository Service, Integration Service, Error Logs, and more. Logs will be saved in a user created directory: *C:\Informatica\10.5.1\logs*

4.1 Fetch All Domain Logs

Captures domain-wide logs.

```
infacmd.bat isp GetLog -dn <DomainName> -un <Username> -pd <Password> -fm <Format> -lo  
<LogFilePath>
```

- <DomainName>: Name of your Informatica domain (e.g., Domain)
- <Username>: Administrator or authorized user
- <Password>: User password
- <Format>: Log format (Text, XML, etc.)
- <LogFilePath>: Full path where the log will be saved

```
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -fm Text -lo  
C:\Informatica\10.5.1\logs\DomainLogs.txt
```

4.2 Fetch Repository Service Logs (REPO_ORACLE)

Captures repository-specific logs.

```
infacmd.bat isp GetLog -dn <DomainName> -un <Username> -pd <Password> -sn <ServiceName> -  
fm <Format> -lo <LogFilePath>
```

- <ServiceName>: Name of the repository service (e.g., REPO_ORACLE)

- Other parameters same as above

```
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -sn REPO_ORACLE -fm Text -lo C:\Informatica\10.5.1\logs\RepositoryServiceLogs.txt
```

4.3 Fetch Integration Service Logs (INT_SERV_ORACLE)

Captures integration service logs.

```
infacmd.bat isp GetLog -dn <DomainName> -un <Username> -pd <Password> -sn <ServiceName> -fm <Format> -lo <LogFilePath>
```

- <ServiceName>: Name of the integration service (e.g., INT_SERV_ORACLE)
- Other parameters same as above

```
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -sn INT_SERV_ORACLE -fm Text -lo C:\Informatica\10.5.1\logs\IntegrationServiceLogs.txt
```

4.4 Fetch All Logs (Full System)

Captures all logs from the domain and services.

```
infacmd.bat isp GetLog -dn <DomainName> -un <Username> -pd <Password> -fm <Format> -lo <LogFilePath>
```

- This command fetches logs across **all services** in the domain.
- No specific service name (-sn) is required.

```
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -fm Text -lo C:\Informatica\10.5.1\logs\FullSystemLogs.txt
```

4.5 Fetch Only Error and Critical Logs

Captures only error and critical logs.

```
infacmd.bat isp GetLog -dn <DomainName> -un <Username> -pd <Password> -svt <Severity> -fm <Format> -lo <LogFilePath>
```

- <Severity>: Filter logs by severity (Error, Warning, Info)
- This example uses Error to fetch only **error and critical** logs.

```
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -svt Error -fm Text -lo C:\Informatica\10.5.1\logs\ErrorLogs.txt
```

4.6 Fetch Logs Within a Specific Timeframe (Last 30 Days)

Captures logs from January 24, 2024, to February 24, 2024.

```
infacmd.bat isp GetLog -dn <DomainName> -un <Username> -pd <Password> -sd "<StartDate>" -ed "<EndDate>" -fm <Format> -lo <LogFilePath>
```

- <StartDate>: Start of the desired log period (e.g., "2024-01-24 00:00:00")
- <EndDate>: End of the desired log period (e.g., "2024-02-24 23:59:59")
- Fetches logs **only within the specified date range**.

```
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -sd "2024-01-24
00:00:00" -ed "2024-02-24 23:59:59" -fm Text -lo
C:\Informatica\10.5.1\logs\Last30DaysLogs.txt
```

4.7 Batch File to Run All Commands at Once

To automate the process, create a batch file named *fetch_all_logs.bat* with the following content:

```
@echo off
echo Fetching Domain Logs...
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -fm Text -lo
C:\Informatica\10.5.1\logs\DomainLogs.txt

echo Fetching Repository Service Logs...
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -sn REPO_ORACLE -fm
Text -lo C:\Informatica\10.5.1\logs\RepositoryServiceLogs.txt

echo Fetching Integration Service Logs...
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -sn INT_SERV_ORACLE -fm
Text -lo C:\Informatica\10.5.1\logs\IntegrationServiceLogs.txt

echo Fetching Full System Logs...
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -fm Text -lo
C:\Informatica\10.5.1\logs\FullSystemLogs.txt

echo Fetching Error Logs Only...
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -svt Error -fm Text -lo
C:\Informatica\10.5.1\logs\ErrorLogs.txt

echo Fetching Logs for the Last 30 Days...
infacmd.bat isp GetLog -dn Domain -un Administrator -pd Admin_123 -sd "2024-01-24
00:00:00" -ed "2024-02-24 23:59:59" -fm Text -lo
C:\Informatica\10.5.1\logs\Last30DaysLogs.txt

echo All logs fetched successfully!
pause
```

4.8 Summary of Output Log Files

Log Type	Output File
Domain Logs	C:\Informatica\10.5.1\logs\DomainLogs.txt
Repository Service Logs	C:\Informatica\10.5.1\logs\RepositoryServiceLogs.txt
Integration Service Logs	C:\Informatica\10.5.1\logs\IntegrationServiceLogs.txt
Full System Logs	C:\Informatica\10.5.1\logs\FullSystemLogs.txt
Error Logs Only	C:\Informatica\10.5.1\logs\ErrorLogs.txt
Last 30 Days Logs	C:\Informatica\10.5.1\logs\Last30DaysLogs.txt