THREADCONNECT MONITORING DRIVEN DEVELOPMENT GUIDE

1. INTRODUCTION

Monitoring includes tracking different statistics of the system like errors, memory usage, CPU usage, Data Flow statistics, etc. **Monitoring Driven Development** (MDD) in ThreadConnect provides the ability to monitor and report the health of the business flows using Reporting tasks. Reporting tasks provide an efficient way to extract and process the data generated by the framework for technical related metrics (system) as well as workflow related metrics (business).

The following table lists down some of the important Reporting tasks offered by ThreadConnect:-

Reporting Task Name	Description
SiteToSiteBulletinReportingTask	To report the errors and warning in bulletins using Site to Site protocol.
SiteToSiteStatusReportingTask	To report ThreadConnect status (component wise) events using Site to Site protocol.
SiteToSiteProvenanceReportingTask	To report the ThreadConnect Data Provenance events using Site to Site protocol.
ControllerStatusReportingTask	Logs the 5-minute stats that are shown in the ThreadConnect Summary Page for Processors and Connections.
MonitorDiskUsage	To report and warn about the disk usage of a specific directory.
MonitorMemory	To monitor the amount of Java Heap used in a Java Memory pool of JVM.

2. MONITORING DRIVEN DEVELOPMENT FLOW DESCRIPTION

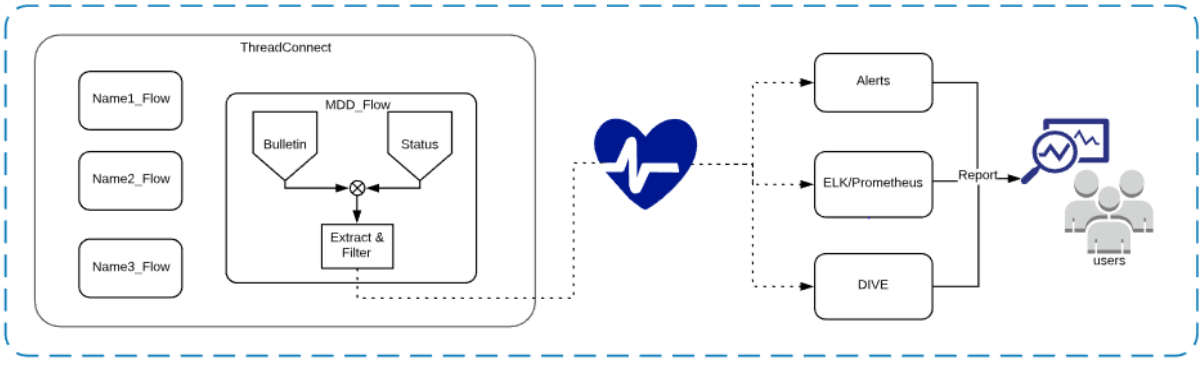


Figure: Design for ThreadConnect' s MDD Flow

The template is designed to capture Flow-Level Status and Errors using SiteToSiteStatusReportingTask and SiteToSiteBulletinReportingTask Reporting tasks respectively thereby making ThreadConnect monitor itself. These records are processed separately based on the type of record (Bulletin or Status) then filtered, and then required fields are extracted in a format which can be routed to either Reporting tools (Email, ServiceNow incident) or Visualization tools (DIVE, Grafana or Kibana) based on the requirement.

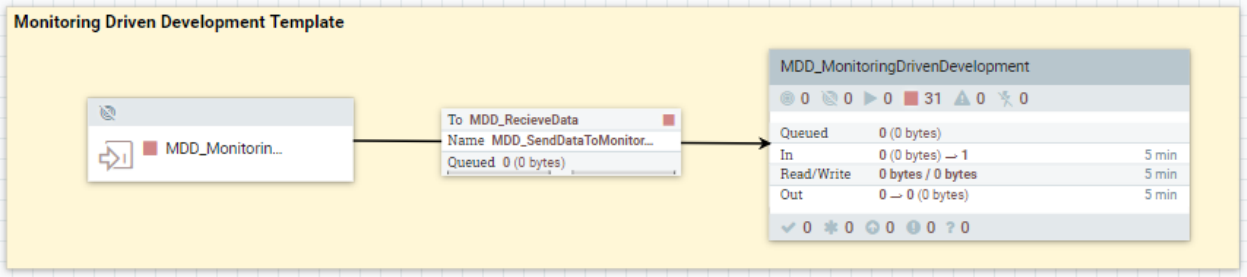


Figure: Monitoring Driven Development Flow

Top level of the Flow:-

- *Input Port* (MDD_Monitoring) – Input source to which the Reporting tasks routes the captured records.
  NOTE: This Input Port is specific to the MDD and should not be used by other components

- *ProcessGroup* (MDD_MonitoringDrivenDevelopment) – Flow to process (filter and extract) the incoming Bulleting and Status records. This ProcessGroup requires the following variables set by the user.

Variable Name	Variable Description
TC.Environment	Specifies the ThreadConnect Environment where the MDD flow is running. (ThreadConnect DEV/STAGE/QA/PROD)

- *SiteToSiteBulletinReportingTask* (MDD_SiteToSiteBulletinReportingTask) – Monitors the ThreadConnect instance for Bulletin messages.

- *SiteToSiteStatusReportingTask* (MDD_SiteToSiteStatusReportingTask) – Monitors the ThreadConnect instance and captures the status of every component (Processors, Ports, Processgroup, Connections, RPG etc) in the canvas.

- *AvroRecordSetWriter* (MDD_AvroRecordSetWriter) – Writes the contents of a RecordSet in Binary Avro format. This controller service is required for both the Reporting tasks.

MDD_MonitoringDrivenDevelopment ProcessGroup level:-

- *Input Port* (MDD_RecieveData) – Receives the incoming flowfiles from the Input port at the top level entering the ProcessGroup.

- *RouteOnAttribute* (MDD_CheckForTypeOfData_RouteOnAttribute) – Checks flowfile attributes for the type of record being transferred by the reporting tasks (either Bulletins or Status), based on which the incoming flowfiles are routed to the downstream flows.

- *ProcessGroup* (MDD_BulletinReportingTask) – Flow to extract necessary fields from records captured by SiteToSiteBulletinReportingTask and converts these extracted fields to the format based on the target's requirement.

- *ProcessGroup* (MDD_StatusReportingTask) – Flow to extract necessary fields from records captured by SiteToSiteStatusReportingTask and converts these extracted fields to the format based on the target's requirement.

- *AvroReader* (MDD_AvroReader) – Parses the incoming flowfile with Avro data and returns each Avro record as a separate Record object. This controller service is used by UpdateRecord processor of the sub processgroups (MDD_StatusReportingTask & MDD_BulletinReportingTask).

- *AvroSchemaRegistry* (MDD_AvroSchemaRegistry) – Registers schemas (JSON format) for both SiteToSiteStatusReportingTask and SiteToSiteBulletinReportingTask.

- *JsonRecordSetWriter* (MDD_JsonRecordSetWriter) – Writes the results of the RecordSet as JSON Array based on the schema specified in MDD_AvroSchemaRegistry controller service. It's used by UpdateRecord and QueryRecord processors of the sub processgroups (MDD_StatusReportingTask & MDD_BulletinReportingTask).

- *JsonTreeReader* (MDD_JsonTreeReader) – Parses JSON object into individual Record objects based on the schema specified in MDD_AvroSchemaRegistry controller service. It's used by QueryRecord processors of the sub processgroups (MDD_StatusReportingTask & MDD_BulletinReportingTask).

OPTIONAL STEPS

- *RouteOnAttribute* (MDD_RouteOnIntegration_RouteOnAttribute) – Routes incoming flowfile based on the Integration Name.

- *PutEmail* (MDD_PutEmail) – Sends out the incoming flowfile as Email body.

- *LogAttribute* (MDD_LogAttribute) – Logs the attributes of the incoming flowfile, once they have been successfully processed by the PutEmail processor.

MDD_BulletinReportingTask ProcessGroup level:-

- *Input Port* (MDD_RecieveBulletinRecords) – Receives the incoming flowfiles containing Bulletin records coming from the parent ProcessGroup.

- *UpdateRecord* (MDD_InitializeProjectNameFieldForBulletins_UpdateRecord) – Updates the contents of the incoming flowfile by adding new field (Integration Name) to the record.

- *QueryRecord* (MDD_FetchError&WarnBulletinRecords_QueryRecord) – Executes SQL queries on the incoming flowfiles and routes them based on the results to the downstream flows. Bulletin messages with log level ERROR and WARNING is extracted at both Root Level (as some components are present only at the Root level of the ThreadConnect canvas) and Integration (business) Level.

- *EvaluateJSONPath* – Evaluates the incoming flowfile containing JSON records and extracts specified fields and writes the value as attributes of outgoing flowfile.

- *ReplaceText* – Replaces the content of the incoming flowfile with user provided text from the extracted values from flowfile attributes (in the template HTML format is being used)

- *Output Port* (MDD_SendBulletinRecords) – Transfers the incoming flowfiles containing processed data to the parent ProcessGroup.

MDD_StatusReportingTask ProcessGroup level:-

- *Input Port* (MDD_RecieveStatusRecords) – Receives the incoming flowfiles containing Status records coming from the parent ProcessGroup.

- *UpdateRecord* (MDD_InitializeProjectNameFieldForStatus_UpdateRecord) – Updates the contents of the incoming flowfile by adding new field (Integration Name) to the record.

- *QueryRecord* (MDD_FetchStatusRecords_QueryRecord) – Executes SQL queries on the incoming flowfiles and routes them based on the results to the downstream flows. Status records is categorized based on the component type present at the Root level and Integration (business) Level.

- *EvaluateJSONPath* – Evaluates the incoming flowfile containing JSON records and extracts specified fields and writes the value as attributes of outgoing flowfile.

- *ReplaceText* – Replaces the content of the incoming flowfile with user provided text from the extracted values from flowfile attributes (in the template HTML format is being used).

- *Output Port* (MDD_SendStatusRecords) – Transfers the incoming flowfiles containing processed data to the parent ProcessGroup.

3. TEMPLATE REQUIREMENTS

1. Drag the template icon, select the MDD_Template from the drop-down option. This would add the flow onto the canvas. Add the following components (Click on the 'Controller Settings' option in the Global menu present in the top right corner as shown below):-

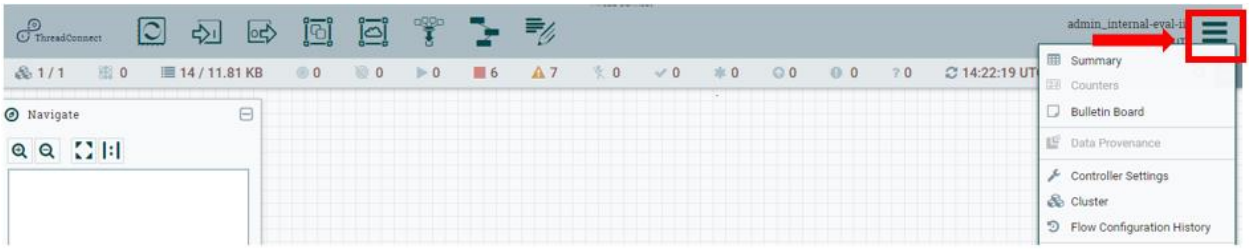
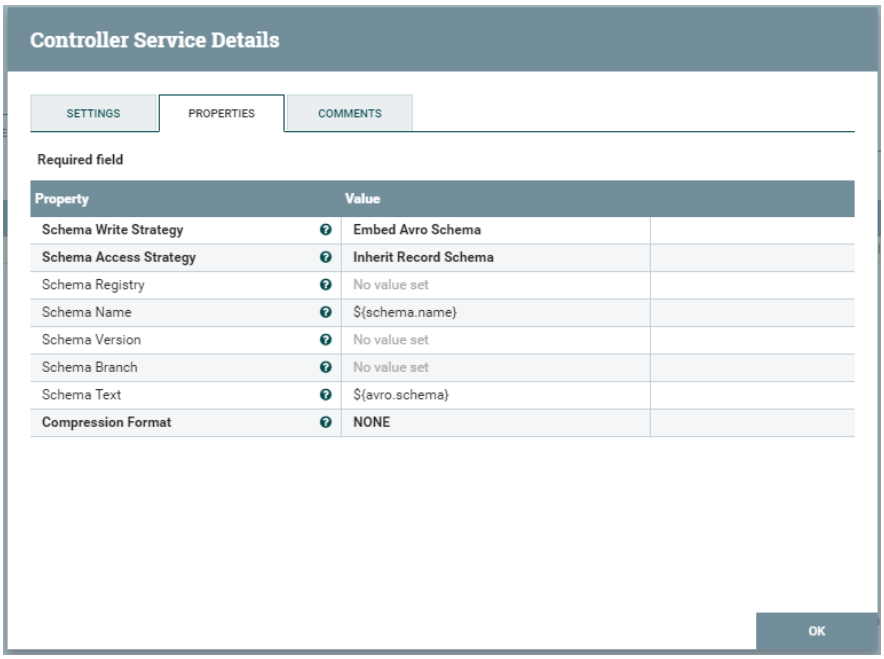
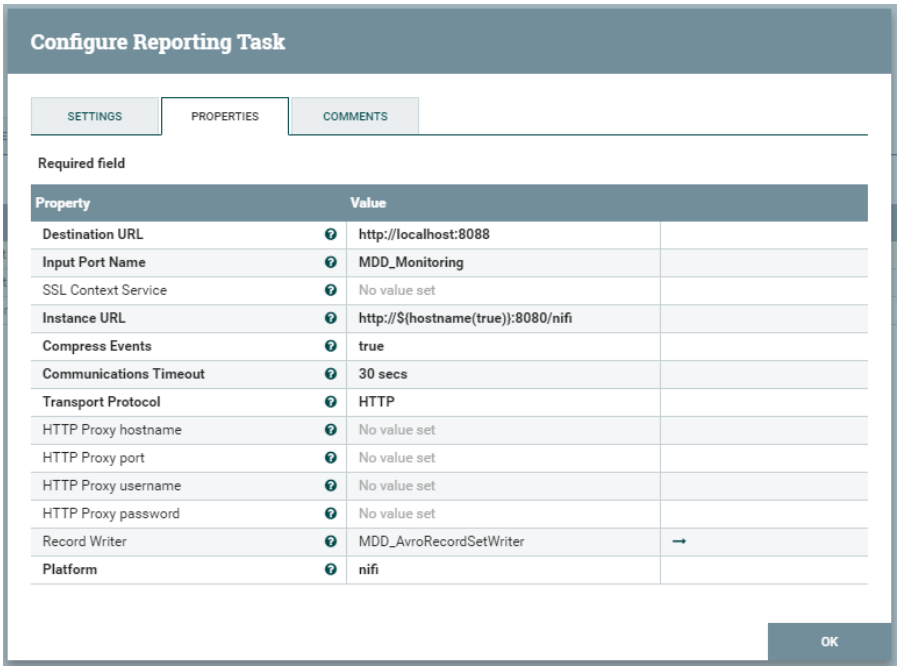


Figure: Global Menu of ThreadConnect canvas

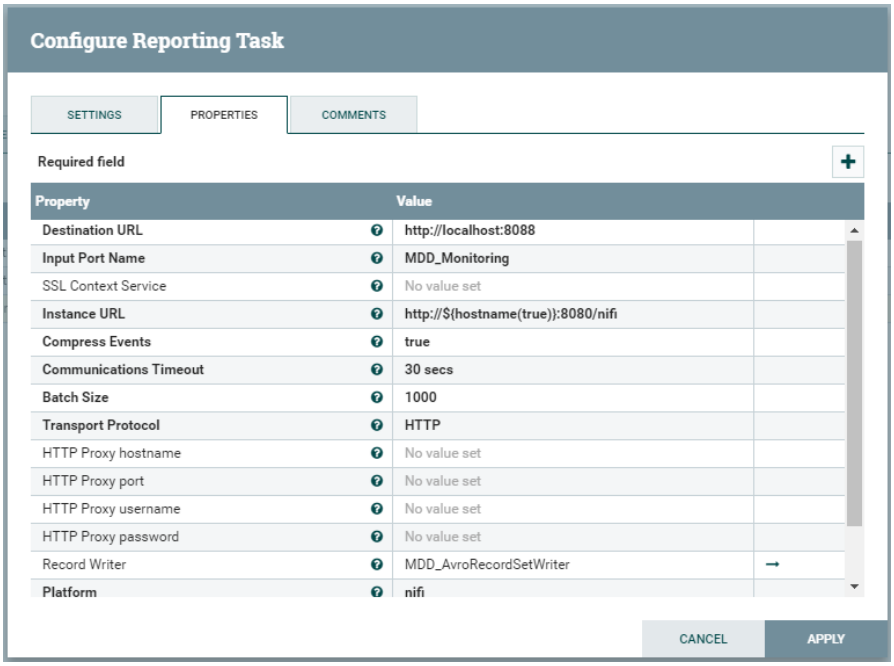
a. AvroRecordSetWriter (Under Reporting Task Controller Services Tab of Controller Settings). Name the controller service as MDD_AvroRecordSetWriter, configure the properties with following values.

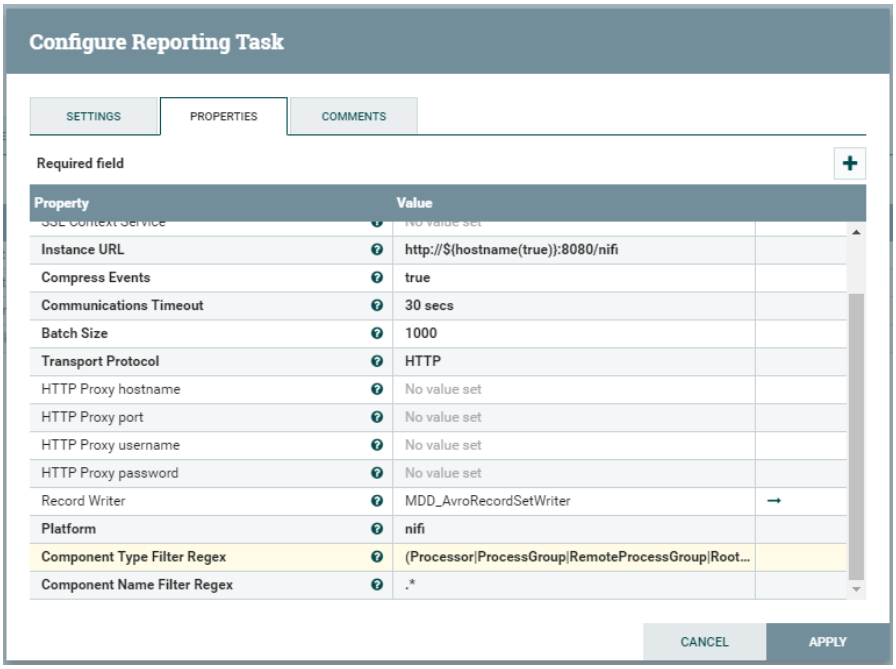


b. SiteToSiteBulletinReportingTask (Under Reporting Tasks Tab of Controller Settings). Name the Reporting task as MDD_SiteToSiteBulletinReportingTask, configure the properties with following values (Based on the ThreadConnect instance where the MDD flow is running, the URL of the instance should be provided as value for Destination URL and Select the MDD_AvroRecordSetWriter controller service which was added in the previous step for Record Writer).



c. SiteToSiteStatusReportingTask (Under Reporting Tasks Tab of Controller Settings). Name the Reporting task as MDD_ SiteToSiteStatusReportingTask, configure the properties with following values (Based on the ThreadConnect instance where the MDD flow is running, the URL of the instance should be provided as value for Destination URL and Select the MDD_AvroRecordSetWriter controller service for Record Writer).





2. Based on the environment in which MDD flow is running, provide necessary value for the TC.Environment variable of MDD_MonitoringDrivenDevelopment ProcessGroup.

3. Based on the requirement, MDD_RouteOnIntegration_RouteOnAttribute and MDD_PutEmail processors of MDD_MonitoringDrivenDevelopment ProcessGroup can be replaced with connection to any visualization tools. Change the HTML text in the Replace Text

processors of MDD_BulletinReportingTask & MDD_StatusReportingTask sub ProcessGroups into the format needed for these tools.

4. Enable all the controller services within the MDD_MonitoringDrivenDevelopment ProcessGroup. Run the MDD flow by selecting the MDD_Monitoring Input Port and MDD_MonitoringDrivenDevelopment ProcessGroup components. Once the flow has started to run, then enable the MDD_AvroRecordSetWriter controller service, MDD_SiteToSiteBulletinReportingTask & MDD_SiteToSiteStatusReportingTask reporting tasks.

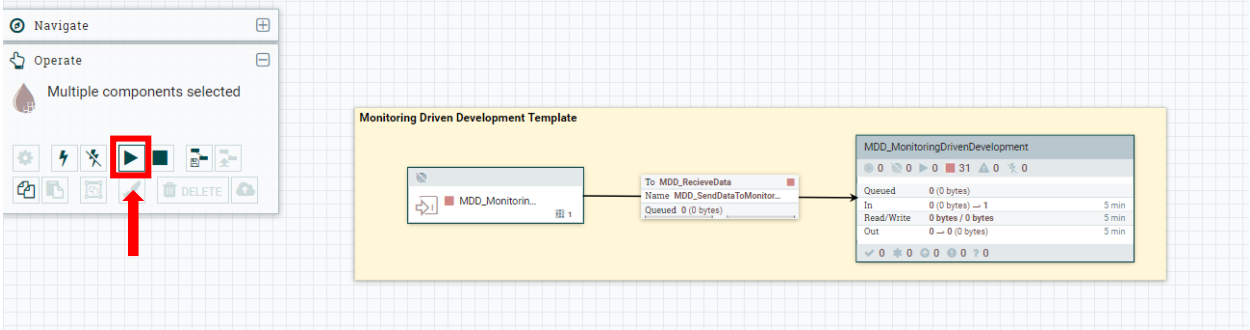


Figure: Enabling the MDD flow on the ThreadConnect canvas

5. The Reporting tasks can be scheduled based on the time interval between the records. By default, its scheduled for 5mins.

6. The MDD_Monitoring Input Port is preserved for monitoring purpose and should not be used by any other components.

7. User when creating a new business flow should follow strict naming conventions for all the components (Processors, Ports, Processgroup, Connections etc) making the flow which is a main requirement for the MDD flow.
   Integration Flow Naming Conventions (E.g. IntegrationName_)

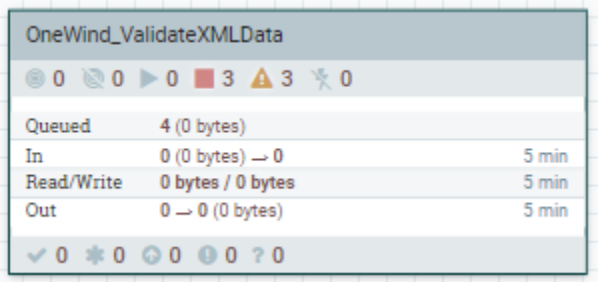


Figure: OneWind Integration Flow

8. OPTIONAL: Based on requirements, additional SQL statements can be provided in the QueryRecord processor to extract more details from Bulletin and Status records.