#pw2025

Power Week 2025

18 - 19 - 20 novembre 2025

IBM Innovation Studio Paris

S66 – MANZAN : Vers une surveillance unifiée des évènements IBM i

20 novembre 11:15 - 12:15

Gautier Dumas
CFD-Innovation
gdumas@cfd-innovation.fr

IBM

common FRANCE

C/D

〈INNOVATION〉

- Conseil
- Formation
- Développement

ASSISTANCE

- Web et Open Source
  - php
  - git
  - open source
  - node.js
- Communication
  - OpenSSL – Cryptography and SSL/TLS Toolkit
  - SOAP / REST – Web Services
  - TOP SECRET
- Valorisation des données
  - SYD
  - SQL
  - IBM DB2

Power Week

18 -19 - 20 novembre 2025

MANZAN

Qu'est-ce que c'est ?

Qu'est ce que Manzan ?

- Manzan est un projet open source visant à fournir une passerelle entre :
  - Des évènements IBM i de tous type
  - Vers une variété de destinations externes
- Afin de centraliser et d'homogénéiser la gestion des évènements IBM i

- Renforce l’intégration de l’IBM i dans les solutions de supervision d’infrastructures hétérogènes.
- Conçu pour simplifier l'exploitation et la surveillance des systèmes IBM i

Qui est derrière Manzan ?

- Jesse Gorzinski (ThePrez)
- Sanjula Ganepola
- Jonnyz32
- Liam Barry Allan (worksofliam)
- …

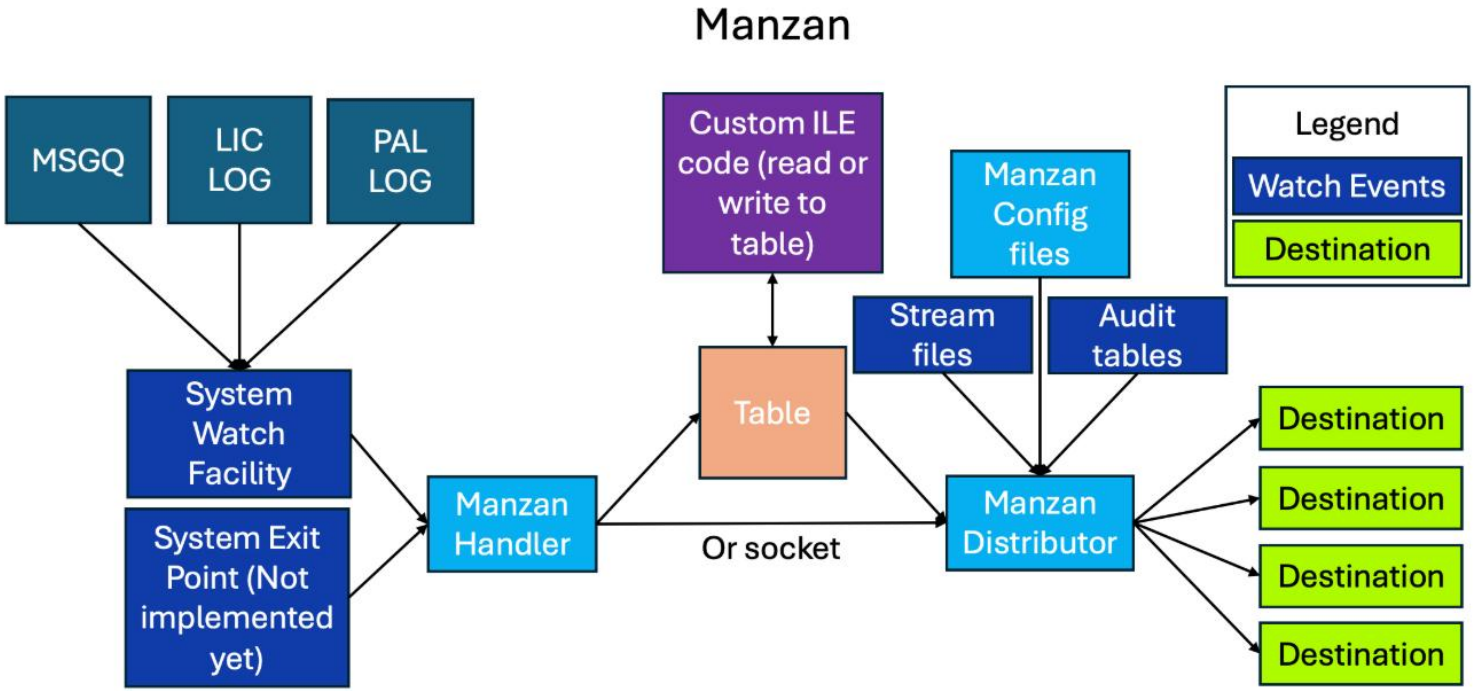
- https://github.com/ThePrez/Manzan/graphs/contributors

Pourquoi le nom Manzan ?

- Doit son nom à un havre de paix qui apporte réconfort à son créateur, Jesse Gorzinski.
- Tout comme ce que la solution doit apporter aux professionnels de l'informatique qui souhaite gérer des tâches complexes

Architecture

- Composants principaux



Gestionnaire et distributeur

- Deux composants principaux :
  - Le gestionnaire (input)
    - Reçoit et gère les évènements
    - Transforme les données en un format exploitable
  - Le distributeur (output)
    - Formate et filtre les données
    - Envoi les données à la destination configurée

Destinations possibles

- Code personnalisé ✅
- ActiveMQ ✅
- AWS Simple Email Service (SES) ⏳
- AWS Simple Notification System (SNS) ⏳
- Elasticsearch ✅
- Email (SMTP/SMTPS) ✅
- FluentD ✅
- Google Drive ⏳
- Google Pub/Sub ✅
- Grafana Loki ✅
- HTTP endpoints (REST, etc) ✅
- HTTPS endpoints (REST, etc) ✅
- Internet of Things (mqtt) ⏳
- Kafka ✅
- Mezmo ✅
- Microsoft Teams ⏳
- PagerDuty ✅
- Sentry ✅
- Slack ✅
- SMS (via Twilio) ✅
- Splunk ✅

✅ = implemented 🌓 = partially implemented ⏳ = future

Power Week

18 -19 - 20 novembre 2025

Manzan

Installation

Prérequis et installeur

- Utiliser SSH
- Java (min version 8)
- Téléchargement de l'installeur (format jar) sur https://github.com/ThePrez/Manzan/releases
  - Dernière release (à date d'écriture du document) : v0.0.14 du 10/07/2025
- Lancement de l'installeur
  - java –jar manzan-installer-v0.0.9.jar

Configuration

- Les configurations se trouvent sous /QOpenSys/etc/manzan
  - Elles servent notamment à configurer
    - Manzan (configuration générale)
    - Les sources
    - Les destinations
  - Détails dans la suite du support

Démarrage de Manzan

- En interactif depuis ssh
  - /opt/manzan/bin/manzan
- A l'aide de service-commander
  - yum install service-commander
  - sc start/check/stop manzan

Manzan

Configuration générale
app.ini

Configuration générale

- fichier /QOpenSys/etc/manzan/app.ini
  - Partie install
    - Définit la librairie qui contient le composant ILE
  - Partie remote
    - Non utilisé quand fonctionnement sur IBM i

```
[install]
library=MANZAN

[remote]
system=youribmi
user=manzanuser
password=yoyoyo
```

app.ini

Logging

- Manzan peut être lancé avec plusieurs niveaux de logs (variable d'environnement)
  - MANZAN_DEBUG_LEVEL = 0: Pas de log
  - MANZAN_DEBUG_LEVEL = 1: Log seulement en cas d'erreurs
  - MANZAN_DEBUG_LEVEL = 2: Log pour les warnings et erreurs
  - MANZAN_DEBUG_LEVEL = 3: Log pour les infos, warnings et erreurs

- *ADDENVVAR ENVVAR(MANZAN_DEBUG_LEVEL) VALUE(3) LEVEL(*SYS) REPLACE(*YES)*
  - *Restart de *SSHD pour prise en compte*

- On retrouvera la log dans /tmp/manzan_debug.txt

Manzan

Configuration des sources
data.ini

Configuration des sources

- fichier /QOpenSys/etc/manzan/data.ini
  - 1 section par source
    - Un type (liste prédéfinie, voir diapo suivante)
    - 1 ou n destinations (à définir dans dests.ini)
    - D'autre(s) propriété(s) en fonction du type de la source

```
[<id>]
# where the data is coming from
type=<type>
# as defined in dests.ini
destinations=<destinations>

# other properties for <id> here..
```

data.ini

Configuration des sources – Liste des types

id	Description	Required properties	Optional properties
`file`	Triggered when a file changes	`file` path of file to watch	* `filter` only listen for lines that include this value
`watch`	Triggered when the Manzan handler is called	`id` of the watch (session ID) `strwch` is part of the STRWCH CL command that can be used to describe how to start the watch when Manzan starts up	* `numToProcess` can be used to configure how many messages are queried for by the distributor (default `1000`) * `interval` the interval in ms at which to query for new messages
`table`	Triggered when data is inserted into the specified table	`table` name of the table to watch `schema` the schema in which the table resides	* `numToProcess` can be used to configure how many messages are queried for by the distributor (default `1000`) * `interval` the interval in ms at which to query for new messages
`audit`	Triggered when the specified `auditType` catches an event.	`auditType` options are `AUTHORITY_FAILURE, AUTHORITY_CHANGES, COMMAND_STRING, CREATE_OBJECT, USER_PROFILE_CHANGES, DELETE_OPERATION, ENVIRONMENT_VARIABLE, GENERIC_RECORD, JOB_CHANGE, OBJECT_MANAGEMENT_CHANGE, OWNERSHIP_CHANGE, PASSWORD, SERVICE_TOOLS_ACTION, ACTION_TO_SYSTEM_VALUE, READ_OF_OBJECT, CHANGE_TO_OBJECT, NETWORK_PASSWORD_ERROR, SYSTEMS_MANAGEMENT_CHANGE, SOCKETS_CONNECTIONS, PRIMARY_GROUP_CHANGE_FOR_RESTORED_OBJECT, OWNERSHIP_CHANGE_FOR_RESTORED_OBJECT, AUTHORITY_CHANGE_FOR_RESTORED_OBJECT, PTF_OBJECT_CHANGE, PROFILE_SWAP, PRIMARY_GROUP_CHANGE, PTF_OPERATIONS, PROGRAM_ADOPT, OBJECT_RESTORE, ATTRIBUTE_CHANGE, DB2_MIRROR_REPLICATION_STATE, DB2_MIRROR_PRODUCT_SERVICES, DB2_MIRROR_REPLICATION_SERVICES, DB2_MIRROR_COMMUNICATION_SERVICES, DB2_MIRROR_SETUP_TOOLS, LINK_UNLINK_SEARCH_DIRECTORY, INTRUSION_MONITOR, SERVICE_TOOLS_USER_ID_AND_ATTRIBUTE_CHANGES, ROW_AND_COLUMN_ACCESS_CONTROL, ATTRIBUTE_CHANGES, ADOPTED_AUTHORITY, AUDITING_CHANGE`	* `fallbackStartTime` is the number of hours prior to the current date that we will query for audit messages, if no audit messages have been queried before * `numToProcess` can be used to configure how many messages are queried for by the distributor (default `1000`) * `interval` the interval in ms at which to query for new messages
`sql`	Executes arbitrary sql at a predefined interval	`query` to be executed	* `interval` the interval in ms at which to run the sql statement
`cmd`	Executes an arbitrary command at a predefined interval	`cmd` command to be executed	* `args` The arguments to be passed to this command. The full command to be executed will be `<cmd> <args>`. Note. Chaining commands together will not work. In the case of needing to execute multiple commands, try putting them in a script and then executing the script. * `interval` the interval in ms at which to run the command.
`http`	Fetches data from an http endpoint at the specified interval	`url` to make an http request from including any path parameters	* `interval` the interval at which to query for new messages * `filter` only listen for responses that include this value * <header> any header key value pair to be used for this http request

Exemple – watch message queue

- Surveillance de la message queue QSYSOPR. Source désactivée.

```
[watcher1]
type=watch
id=QSYSOPR
destinations=test_out, slackme
strwch=WCHMSG((*ALL)) WCHMSGQ((QSYS/QSYSOPR))
enabled=false
```

Exemple – watch *HSTLOG

- Gestion des messages de HSTLOG avec formatage des valeurs. Démarre avec Manzan. Lecture de 10000 messages toutes les 10 millisecondes

```
[watchout]
type=watch
id=jesse
destinations=test_out, slackme
format=$MESSAGE_ID$ (severity $SEVERITY$): $MESSAGE$
strwch=WCHMSG((*ALL)) WCHMSGQ((*HSTLOG))
interval=10
numToProcess=10000
```

Exemple - file

- Surveillance d'un fichier de l'IFS, si la nouvelle ligne contient "error"

```
[logfile1]
type=file
file=test.txt
destinations=email_it, test_out
filter=error
format=$FILE_DATA$
```

Exemple - table

- Ecriture d'un message dans slack et kafka quand une donnée est écrite dans la table foo.announcements

```
[tab1]
type=table
table=announcements
schema=foo
destinations=slack, kafka
format=$FIELD1$, $FIELD2$
```

Exemple - audit

- Déclenchement d'un message dans Slack dès tentative de connexion infructueuse sur le système
  - S'appuie sur les Audit journal entry services (nécessite l'activation de l'audit journaling): https://www.ibm.com/docs/fr/i/7.6.0?topic=services-audit-journal-entry

```
[audit1]
type=audit
destinations=slackme
auditType=PASSWORD
fallbackStartTime=1
format=Violation type: $VIOLATION_TYPE_DETAIL$, username: $AUDIT_USER_NAME$
hostname: $HOSTNAME$ timezone:$TIMEZONE$
interval=5000
injections.HOSTNAME="myhost@abc.com"
injections.TIMEZONE="EST"
```

Type audit et auditType

- auditType possibles :
  - AUTHORITY_FAILURE,
  - AUTHORITY_CHANGES,
  - COMMAND_STRING,
  - CREATE_OBJECT,
  - USER_PROFILE_CHANGES,
  - DELETE_OPERATION,
  - ENVIRONMENT_VARIABLE,
  - GENERIC_RECORD,
  - JOB_CHANGE,
  - OBJECT_MANAGEMENT_CHANGE,
  - OWNERSHIP_CHANGE,
  - PASSWORD,
  - SERVICE_TOOLS_ACTION,
  - ACTION_TO_SYSTEM_VALUE,
  - READ_OF_OBJECT,
  - CHANGE_TO_OBJECT,
  - NETWORK_PASSWORD_ERROR,
  - SYSTEMS_MANAGEMENT_CHANGE,
  - SOCKETS_CONNECTIONS,
  - PRIMARY_GROUP_CHANGE_FOR_RESTORED_OBJECT,
  - OWNERSHIP_CHANGE_FOR_RESTORED_OBJECT,
  - AUTHORITY_CHANGE_FOR_RESTORED_OBJECT,
  - PTF_OBJECT_CHANGE,
  - PROFILE_SWAP,
  - PRIMARY_GROUP_CHANGE,
  - PTF_OPERATIONS,
  - PROGRAM_ADOPT,
  - OBJECT_RESTORE,
  - ATTRIBUTE_CHANGE,
  - DB2_MIRROR_REPLICATION_STATE,
  - DB2_MIRROR_PRODUCT_SERVICES,
  - DB2_MIRROR_REPLICATION_SERVICES,
  - DB2_MIRROR_COMMUNICATION_SERVICES,
  - DB2_MIRROR_SETUP_TOOLS,
  - LINK_UNLINK_SEARCH_DIRECTORY,
  - INTRUSION_MONITOR,
  - SERVICE_TOOLS_USER_ID_AND_ATTRIBUTE_CHANGES,
  - ROW_AND_COLUMN_ACCESS_CONTROL,
  - ATTRIBUTE_CHANGES,
  - ADOPTED_AUTHORITY,
  - AUDITING_CHANGE

Exemple - sql

- Obtenir les dernières données de la table sample.department toutes les 5 minutes

```
[sql1]
type=sql
query=SELECT * FROM sample.department WHERE time > CURRENT_TIMESTAMP - 5 MINUTES
destinations=googpubsub, stdout
format=department: $DEPTNAME$
interval=300000
```

Exemple - cmd

- Afficher la job log d'un job toutes les minutes

```
[cmd1]
type=cmd
destinations=stdout
cmd=system
args="DSPJOBLOG JOB(047284/QTMHHTTP/ADMIN)"
format=cmd $CMD$ args $ARGS$ exitval $EXITVALUE$ stderr $STDERR$
stdout $STDOUT$
interval=60000
```

Exemple - http

- Lecture de données provenant d'une url http toutes les 5 minutes.

```
[http1]
type=http
destinations=test_out
format=Result: $results.name.first$ $results.name.last$ $json:results.location$
url=https://fakeusergenerator.com
interval=300000
authorization=bearer xxxxxx
```

Formatage des données

- Dépend de chaque type de source de données
- S'utilise dans la propriété format
- Identification des variables à l'aide des symboles $
  - *Exemple : $VIOLATION_TYPE_DETAILS$*
- Quelques exemples de variables en fonction du type
  - sql : <nom_de_la_colonne>
  - cmd : CMD - $ARGS$ - $EXITVALUE$ - $STDERR$ - $STDOUT$
  - file : $FILE_DATA$ - $FILE_NAME$ - $FILE_PATH$
  - watch msgq : $MESSAGE_ID$ - $SEVERITY$ - JOB - $MESSAGE$ …

- Liste complète sur : https://theprez.github.io/Manzan/#/config/format

Manzan

Configuration des destinations
dests.ini

Configuration des destinations

- fichier /QOpenSys/etc/manzan/ dests.ini
  - 1 section par destination (endroit où les données sont envoyées)
    - Un type (liste prédéfinie, voir diapo suivante)
    - D'autre(s) propriété(s) en fonction du type de la source

```
[<id>]
# the type of destination the data is
going to
type=<type>

# other properties for <id> here..
```

dests.ini

Configuration des destinations – Liste des types

id	Description	Required properties	Commonly used properties	All properties
stdout	Write all data to standard out.	None.		https://camel.apache.org/components/3.22.x/stream-component.html
slack	Send data to a Slack channel	* webhook * channel		https://camel.apache.org/components/3.22.x/slack-component.html
fluentd	Send data to FluentD	* tag * host * port		
file	Send data to a file	* file		https://camel.apache.org/components/3.22.x/stream-component.html
dir	Send data to a directory	* dir		https://camel.apache.org/components/3.22.x/file-component.html
smtp/smtps	Sent data via email	* server * subject * to * from	* port	https://camel.apache.org/components/3.22.x/mail-component.html
sentry	Send data into Sentry	* dsn		
twilio	Send via SMS	* sid * token * to * from		https://camel.apache.org/components/3.22.x/twilio-component.html
loki	Send data into Grafana Loki	* url * username * password	* maxLabels The maximum number of indexed labels each log record can have. The default is 15. If you want to use more than 15 labels, you must also configure this separately on your Grafana Loki instance. However, this is not recommended as it will degrade performance. * labels specifies the labels to be associated with each log record. Each label should be delimited by a space. I.e labels=REMOTE_PORT SYSTEM_NAME (In this case REMOTE_PORT and SYSTEM_NAME should be valid format options on the data source. For example if using Audit Password events). It is strongly recommended to specify this property, because the default option is to add labels for each data attribute that is received.	

Configuration des destinations – Liste des types

id	Description	Required properties	Commonly used properties	All properties
`http/https`	Send data via http/https	* `url`	* `httpMethod` * x where x is any query parameter	https://camel.apache.org/components/3.22.x/http-component.html
`activemq`	Send data to ActiveMQ	* `destinationName`	* `destinationType` * `brokerURL`	https://camel.apache.org/components/3.22.x/activemq-component.html
`splunk-hec`	Send data to Splunk	* `splunkUrl` * `token` * `index`	* `skipTlsVerify`	https://camel.apache.org/components/3.22.x/splunk-hec-component.html
`pagerduty`	Send data to PagerDuty	* `routingKey`	* `component` * `group` * `class`	
`mezmo`	Send data to Mezmo	* `apiKey`	* `tags` * `app`	
`elasticsearch`	Send data to Elasticsearch	* `endpoint` * `apiKey` * `index`		
`otlp`	Send data to OpenTelemetry server	* `url`	* `errorRegex`	

Exemple - smtps

- Destination mail

```
[email_b]
type=smtps
format=Hey, check out this information!! \n\n$FILE_DATA$
  server = my.smtpserver.com
  subject = Testemail
  from=me@mycompany.com
  to=me@mycompany.com
```

Exemple - stdout

- Affichage simple

```
[test_out]
type=stdout
```

Exemple – twilio (sms)

- Pour envoi d'un sms via la plateforme twilio

```
[twilio_sms]
type=twilio
componentOptions.sid=x
componentOptions.token=x
to=+x
from=+x
```

Exemple –loki

- Vers la base de données de logs Grafana Loki

```
[loki_out]
type=loki
url=<loki_url>
username=<loki_username>
password=<loki_password>
labels=REMOTE_PORT SYSTEM_NAME
```

Exemple – slack

- Vers la messagerie d'équipe Slack

```
[slackme]
type=slack
channel=open-source-system-status
webhook=https://hooks.slack.com/services/TA3EF58G4...
```

Exemple – http

- Vers un endpoint personnalisé http

```
[myLocalHttpServer]
type=http
url=http://localhost:3000
a=54
b=heybuddy
httpMethod=POST
format={"message": "$FILE_DATA$"}
```

Exemple – https

- Vers un endpoint personnalisé https

```
[myProdServer]
type=https
url=https://production.com
foo=bar
httpMethod=POST
format={"message": "$FILE_DATA$", "path": "$FILE_PATH$",
"name": "$FILE_NAME$"}
```

Exemple - activemq

- Vers une active mq

```
[mq]
type=activemq
destType=queue
destName=TEST.QUEUE
format=This is the $FILE_DATA$
componentOptions.brokerURL=tcp://myactivemq:61616
```

Exemple - splunk

- Vers le SIEM Splunk

```
[splunk_out]
type=splunk-hec
splunkUrl=<splunk_host>:<splunk_port>
token=<splunk_token>
index=<splunk_index>
```

Exemple - elasticsearch

- Vers elasticsearch

```
[elasticsearch_out]
type=elasticsearch
endpoint=<endpoint>
apiKey=<api_key>
index=manzan
```

Démo Manzan

Sources	Destinations
Fichier de l'IFS php_error.log	Console
MSGQ QSYSOPR	Mail
Requête SQL	Sms via twilio
Type table (utiliser comme file)	Slack
	Loki
	http

IBM
common
FRANCE

Power Week

18 -19 - 20 novembre 2025

Manzan

Contribuer à Manzan

Contribuer à Manzan

- En testant la solution dans vos environnements et remontant les bogues rencontrés
  - Avec un titre clair et descriptif
  - Les étapes pour reproduire le problème
  - Le comportement attendu et celui réellement obtenu
  - Toute pièce nécessaire à la compréhension (logs, screenshots, messages d'erreur…)
- En suggérant de nouvelles fonctionnalités ou des évolutions
  - Avec le problème ou la limitation que ça réglerait
  - Une description la plus détaillée possible de la solution proposée
  - D'alternatives considérées si c'est applicable
- En contribuant au code
  - Ouvert à tous à l'aide de PR (Pull Request)
    - Fork du dépôt
    - Création d'une nouvelle branche feature
    - Développement de la fonctionnalité
    - Commit et Push des modifications avec des messages clairs et concis
    - Soumettre la PR
- En contribuant à l'amélioration de la documentation
  - En suivant le même process qu'une contribution de code

- Infos sur : https://theprez.github.io/Manzan/#/contributing?id=how-to-contribute

Power Week

18 -19 - 20 novembre 2025

Manzan

Conclusion

En conclusion

- **Simplicité**
  - Installation rapide
  - Configuration à l'aide de fichiers .ini
- **Beaucoup de sources possibles**
  - Préconfigurées
  - Source personnalisée (sql, cmd, http…)
- **Beaucoup de destinations possibles**
  - Préconfigurées
  - Destination personnalisée (http)


MERCI