ICON IPF Documentation - Reference

Complete section documentation with all subsections

Total subsections: 4

Total pages: 64

Table of Contents

Releases

61 pages in this subsection

Releases

Source: https://docs.ipfdev.co.uk/reference/current/release/releases.html

Releases

Release notes for recent IPF releases are found here. Please explore each release for detailed notes and links to other relevant sections within the documentation.

- IPF-2025.1.0
    - IPF-2025.1.0 - Release Notes
        - Optional Modules - Changes & Fixes
        - Business Functions - Changes & Fixes
        - Core - Changes & Fixes
        - CSM Reachability - Changes & Fixes
        - Bulker & Debulker - Changes & Fixes
        - ODS - Changes & Fixes
        - Operational Dashboard - Changes & Fixes
    - Migration Steps for IPF-2025.1.0
    - Fix Releases
        - IPF-2025.1.2 - Rulebook Release Notes
            - Optional Modules - Changes & Fixes
- IPF-2024.4.0
    - IPF-2024.4.0 - Release Notes
        - Optional Modules - Changes & Fixes
        - CSM Reachability - Changes & Fixes
        - Bulker & Debulker - Changes & Fixes
        - File Poller - Changes & Fixes
        - ODS & GUI - Changes & Fixes
    - Migration Steps for IPF-2024.4.0
- IPF-2024.3.0
    - IPF-2024.3.0 - Release Notes
        - Core - Improvements, Changes & Fixes
        - Optional Modules - Changes & Fixes
        - Bulker & Debulker - Changes & Fixes
        - CSM Reachability - Changes & Fixes
        - Dynamic Processing Settings v2 - Changes & Fixes
        - ODS & GUI - Changes & Fixes
    - Migration Steps for IPF-2024.3.0
- IPF-2024.2.0
    - IPF-2024.2.0 - Release Notes
        - Core - Improvements, Changes & Fixes
        - Optional Modules - Changes & Fixes
        - Bulker & Debulker - Changes & Fixes
        - CSM Reachability - Changes & Fixes
        - ODS & GUI - Changes & Fixes

        - SEPA CT CSM - Changes & Fixes
    - Migration Steps for IPF-2024.2.0
    - Fix Releases
        - IPF-2024.2.1- Release Notes
- IPF-2024.1.0
    - IPF-2024.1.0 - Release Notes
        - Optional Modules - Changes & Fixes
        - Bulker & Debulker - Changes & Fixes
        - CSM Reachability - Changes & Fixes
        - ODS & GUI - Changes & Fixes
        - SEPA CT CSM - Changes & Fixes
    - Migration Steps for IPF-2024.1.0
    - Fix Releases
        - IPF-2024.1.2 - Release Notes
            - Migration Steps for IPF-2024.1.2
        - IPF-2024.1.3 - Release Notes
            - Migration Steps for IPF-2024.1.3
- IPF-2023.4.0
    - IPF-2023.4.0 - Release Notes
        - Migration Steps for IPF-2023.4.0
    - Fix Releases
        - IPF-2023.4.1 - Fix Release Notes
            - Migration Steps for IPF-2023.4.1
- IPF-2023.3.0
    - IPF-2023.3.0 - Release Notes
        - Migration Steps for IPF-2023.3.0
    - Fix Releases
        - IPF-2023.3.0.1 - Fix Release Notes
            - Migration Steps for IPF-2023.3.0.1
- IPF-2023.2.0
    - IPF-2023.2.0 - Release Notes
        - Migration Steps for IPF-2023.2.0
    - Fix Releases
        - IPF-2023.2.0.1 - Fix Release Notes
            - Migration Steps for IPF-2023.2.0.1
        - IPF-2023.2.0.2 - Fix Release Notes
            - Migration Steps for IPF-2023.2.0.2

Release Notes for IPF-2025.1.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2025-1-0/release-IPF-2025-1-0.html

Release Notes for IPF-2025.1.0

This page details everything required to get going on IPF Release 2025.1.0 made available on Thursday 22nd May

> If a new IPF app is generated using the scaffolder pointing at ipf 2025.1.0 and if the optior `-DuseBuildScripts=y` is selected, Maven may report an error during the build process.
>
> To resolve this failure, open up the MPS module of the application in MPS, click migrate when prompted and rebuild (the user should be prompted to rebuild as part of the migration).
>
> This build issue will be fixed in IPF 2025.2.0

Change Spotlight

- Springboot versions uplifted to `3.4.5`
- The ability to use ipf-common-starter-core dependency and disable Akka Clustering with the following properties set (*PAY-12974*):
    - akka.management.enabled = false
    - cluster.bootstrap.enabled = false
    - akka.actor.provider = local
    - management.health.akka-cluster.enabled = false
- ActionTimeOut & ActionRetry Exception Framework Enhancement (*PAY-11898*)
- Akka Discovery MongoDB plugin to persist/read config from DB (*PAY-13018*)
- Debtor Address validations for TIPS & RT1 in payments involving non-EEA countries (*PAY-11593*)
- Determine processingEntity from payment identifier (*PAY-11376*)
- Improve out of the box Grafana dashboards and treat them as production deliverables (*PAY-11067*)
- Improve role based access to IPF components (*PAY-11237*)
- Include Summary Object in Archive Bundle (*PAY-12884*)
- ISO20022 Messages for Orderbook Management (*PAY-11720*)
- JWT Token Signing Support for ES/RS (asymmetric keys) (*PAY-12419*)
- Protect IPF Performance During Future Dated Bulk Executions (*PAY-11053*)
- Providing consistent search function for payments on ODS Inquiry (*PAY-12753*)
- Support accessing Parent Unit of Work Data Structures from child flows (*PAY-11819*)
- Support 'Delta' option for market data ingest as well as full load (*PAY-12352*)
- The Payment Warehouse: a suite of artefacts intended to be used in a 'Scheduled Payments' solution
    - includes HTTP API and client library
- The Payment Releaser: a suite of artefacts intended to be used in a 'Scheduled Payments' solution
- VoP Requester MVP (*PAY-11901*)
- VoP Responder MVP (*PAY-11900*)

Fix Spotlight

- When adding PDS entries for alternative identifiers, nulls are handled gracefully and other alternative identifiers are processed successfully to the payment summary(*PAY-12586*)
- Inbound payments getting rejected at TIPS when TtlIntrBkSttlmAmt and IntrBkSttlmAmt are numercally equal but differs in format
- When creating HttpConnectorTransport treatErrorResponseAsFailureWhen predicate is not passed to the transport
- Persistent/Quartz Scheduler does not work correctly with USER_DEFINED backoff-type
- CSM reachability check failing for a participant added the same day

- Errors thrown in Input Behaviour even though the custom business data are handled in the Input Enrichers
- ActionTimeoutEvent's don't hold data
- Insecure JSON Web Token (JWT) Configuration
- Operational Dashboard
    - The Search Filter are not working for System Events & Message Logs on the Operational Dashboard
    - UI does not expire session or cookie without page refresh
    - The search is character / case sensitive for fields"Rule Name" and "BIC"
    - Audit screen does not show up the results if user searches with ID
    - Recall search screen is not showing the payment status and Reason codes for searched payments on Dashboard (like PACS08)

Breaking Changes

- The SchedulingModuleInterface scheduleJob has changed from a return type of CompletionStage<Void> to CompletionStage<JobSpecificationDto> to align with the new scheduler api
- Configuration parameters for debulker stale component remover have been renamed
- Business metric names are now prefixed with `ipf_` by default, and this cannot be changed. e.g. the metric `businessmetrics_payments_finished` is now `ipf_businessmetrics_payments_finished`.
- The global status config has been changed from an array of strings to an array of objects.
- IPF Archiver can no longer be configured to export V1 bundles, and can only export V2 archive bundles.
    - Please check the individual pages for this release for any breaking changes in specific components for more details and those changes not covered here

Release Details

Binaries

The core binaries are available from IPF-Releases.

Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Akka/Alpakka binaries from 2.9.x onwards are no longer on Maven Central and are available from the Akka Library Repository which Icon mirror.

To enable the new mirror, add the following to your settings.xml in the `<repositories>` section:

```
 <repository>
  <id>akka-repository</id>
  <name>Akka library repository</name>
  <url>https://nexus.ipf.iconsolutions.com/repository/akka-repository</url>
</repository>
```

Some MPS dependencies are also mirrored by Icon and can be found by adding the following repository to the same section:

```
 <repository>
  <id>icon-thirdparty</id>
  <name>IPF icon-thirdparty repository</name>
  <url>https://nexus.ipf.iconsolutions.com/repository/icon-thirdparty</url>
</repository>
```

Add the following to your `<servers>` section for each new repository:

```
 <server>
  <id>Repository name here</id>
  <username>xxxx</username>
  <password>xxxx</password>
</server>
```

Where `xxxx` is the username and password repectively provided to you by Icon.

Developer App

The latest version of the Developer App is **2.4.5**

Scaffolder

The latest version of the Scaffolder is **1.2.5**

2025.1.0 Jar/Pom list

The Jars and Poms for 2025.1.0 and the associated versions are listed here: 2025-1-0-artifacts.xlsx

2025.1.0 Dependency List

The 3rd party dependencies, and their current verions are listed here

---

Optional Modules - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2025-1-0/release-IPF-2025-1-0-aom.html

Optional Modules - Changes & Fixes

This page covers the optional module changes and fixes provided in IPF Release 2025.1.0

Scheme Packs

TIPS CSM

New

- Added new configuration options for enabling or disabling non-EEA debtor address validations for a pacs.008 message with the following defaults (*PAY-12807*):
    - ipf.csm.tips.validation.pacs008.non-eea-debtor-address-validation-inbound.enabled = false
    - ipf.csm.tips.validation.pacs008.non-eea-debtor-address-validation-outbound.enabled = false
- The non-EEA debtor address validation check for pacs.008 messages has been introduced for both outbound (*PAY-12808*) and inbound (*PAY-12809*), provided the associated configuration option is enabled. To enable this functionality, set:
    - **For outbound messages**: ipf.csm.tips.validation.pacs008.non-eea-debtor-address-validation-outbound.enabled = true
    - **For inbound messages**: ipf.csm.tips.validation.pacs008.non-eea-debtor-address-validation-inbound.enabled = true

Fixed

- pacs.028 now correctly contains original pacs.008 MsgId in OrgnlMsgId tag (*PAY-13310*)
- Reviewed/corrected Validations documentation for TIPS (*PAY-13912*)

SIC CSM

Fixed

- Remittance information validation on `pacs.008` is now applied as per scheme rules for inbound and outbound (*PAY-1484*)
- NullPointerException on structured remittance information during validation (*PAY-13578*)
- pacs.028 now correctly contains original pacs.008 MsgId in OrgnlMsgId tag (*PAY-13310*)

New

- When setting `additionalInfo` using the SIC simulator, you can now use SpEL to enhance the generated data (*PAY-1484*)

RT1 CSM

New

- Added new configuration options for enabling or disabling non-EEA debtor address validations for a pacs.008 message with the following defaults (*PAY-12812*):
    - ipf.csm.rt1.validation.pacs008.non-eea-debtor-address-validation-inbound.enabled = false
    - ipf.csm.rt1.validation.pacs008.non-eea-debtor-address-validation-outbound.enabled = false
- The non-EEA debtor address validation check for pacs.008 messages has been introduced for both outbound (*PAY-12813*) and inbound (*PAY-12814*), provided the associated configuration option is enabled. To enable this functionality, set:
    - **For outbound messages**: ipf.csm.rt1.validation.pacs008.non-eea-debtor-address-validation-outbound.enabled = true
    - **For inbound messages**: ipf.csm.rt1.validation.pacs008.non-eea-debtor-address-validation-inbound.enabled = true

Fixed

- pacs.028 now correctly contains original pacs.008 MsgId in OrgnlMsgId tag (*PAY-13310*)

IPF Archiver

Breaking Changes

- IPF Archiver can no longer be configured to export V1 bundles, and can only export V2 archive bundles.

New

- Added the option for a summary to be added to the DataEnvelope when archiving as a custom object representation, default value:
    - *ipf.archiver.bundle.include-summary=false*

Business Metrics

Breaking Changes

- Business metric names are now prefixed with `ipf_` by default, and this cannot be changed. e.g. the metric `businessmetrics_payments_finished` is now `ipf_businessmetrics_payments_finished`. This was previously opt-in with the configuration `ipf.business-metrics-processor.opt-in-to-ipf-prefixed-metric-names = true`. This configuration has also been removed.

Verification of Payee Requester

New

- Delivery of MVP requester application with the following highlights:
    - Integration with FPAD
    - Kafka message logging
    - Validation

Verification of Payee Responder

New

- Delivery of MVP responder application with the following highlights:
    - Integration with FPAD
    - Kafka message logging
    - Validation
    - Blocked phishing attempts

Business Functions - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2025-1-0/release-IPF-2025-1-0-business-functions.html

Business Functions - Changes & Fixes

This page covers changes and fixes to Business Functions provided in IPF Release 2025.1.0

Debulker

New

- New reason codes added to floclient to provide more specific error details
    - IPFDBK004 *File operation failed*
    - IPFDBK005 *Component processing initiation failed*
    - IPFDBK006 *Component store operation failed*
    - IPFDBK007 *Check duplicate transaction cache operation failed*

For the complete list of reason codes, see: Reason Codes

Core - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2025-1-0/release-IPF-2025-1-0-core.html

Core - Changes & Fixes

This page details everything required to get going on IPF Release 2025.1.0

New

- The Payment Warehouse: a suite of artefacts intended to be used in a 'Scheduled Payments' solution
    - includes HTTP API and client library
- The Payment Releaser: a suite of artefacts intended to be used in a 'Scheduled Payments' solution
    - including HTTP API and client library
- Updates to IPF Persistent Scheduler to include a HTTP API and client library
- Documentation for getting started with 'Scheduled Payments' that outlines a build solution including the Payment Warehouse, Payment Releaser and IPF Persistent Scheduler. See Payment Solutions.

Bank Filtering

Changed

- Searchable fields: ruleId, ruleName, csmAgentId and bic now use case insensitive search.

CSM Reachability - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2025-1-0/release-IPF-2025-1-0-csm-reach.html

CSM Reachability - Changes & Fixes

This page covers the changes and fixes provided to CSM Reachability IPF Release 2025.1.0

CSM Reachability API

New

- Added new Domain API endpoint for determining the processing entity responsible for the payment messages they handle.
    - `/v2/determine-processing-entity`
- Added new configuration parameter for connector configuration
    - `ipf.csm-reachability-api.connector.determine-processing-entity`

CSM Reachability

Changed

- Changed implementation so country code is made optional for identifierType = NCC for the following endpoints (before it was conditionally mandatory when identifierType = NCC)
    - /v2/party-entities
    - /v2/settlement-agents
    - /v2/select-csm-agents
    - /v2/validate-csm-reachability
    - /v2/validate-intra-entity-reachability
- Search by and match by `BIC` identifier is now changed to also match head office BICs and 8 character BICs.
    - Head office BIC for an entity is usually an 8 character BIC followed by "XXX". It is possible that sometimes the payments or industry data may have this BIC presented as an 8 character BIC only while the data is held with "XXX" at the end (and vice versa). The BIC matching on reachability implementations now supports these scenarios.
    - This change affects:
        - Endpoints `/determine-processing-entity` , `/settlement-agents` , `/validate-csm-reachability` and `/validate-intra-entity-reachability` .
        - Search for `AgentSettings` , `Processing Entity` , `Intra Entity Parties` and `CSM Participant` settings.

Data Ingestion

New

- Added option to process DELTA Bank Directory Plus files on csm data ingestion service
- Bank Directory Plus file has the following format: BANKDIRECTORYPLUS_XX_DELTA_YYYYMMDD.XML or BANKDIRECTORYPLUS_XX_FULL_YYYYMMDD.XML where XX is version number (currently V3)
- Added option to process DELTA Tips Participant files on csm data ingestion service
- TIPS Directory file has the following format: TIPSXXXDLTAYYYYMMDD.XML or TIPSXXXFULLYYYYMMDD.XML where XXX is alphabetic text

---

Bulker & Debulker - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2025-1-0/release-IPF-2025-1-0-debulker.html

Bulker & Debulker - Changes & Fixes

This page covers the Bulker & Debulker module changes and fixes provided in release IPF-2025.1.0.

Debulker

New

- Whenever a system error event is raised, the reason code and reason description fields on this error are now populated
- New system events raised to provide more specific error information
    - FileOperationFailed
    - ComponentProcessingInitiationFailed
    - ComponentStoreOperationFailed
    - TransactionCacheOperationFailed

For additional details on these newly added system events, see: Debulker System Events

- LocalFileWriter can be configured to overwrite existing files by changing the setting `ipf.file-manager.local.writer.overwrite-existing-file`. It defaults to false, since this was the previous behaviour.

Changed

Breaking Changes

- Name of configuration parameters for `stale component remover` renamed

Stale component remover should be configured using the following config parameters:

```
 ipf.debulker.housekeeping.stale-component-remover.enabled
ipf.debulker.housekeeping.stale-component-remover.initial-delay
ipf.debulker.housekeeping.stale-component-remover.component-max-age
ipf.debulker.housekeeping.stale-component-remover.housekeeping-interval
ipf.debulker.housekeeping.stale-component-remover.scheduler-restart-settings.min-backoff
ipf.debulker.housekeeping.stale-component-remover.scheduler-restart-settings.max-backoff
ipf.debulker.housekeeping.stale-component-remover.scheduler-restart-settings.jitter-factor
```

Removed the following configuration parameters:

```
 ipf.debulker.housekeeping.component.max-age
ipf.debulker.housekeeping.housekeeping-interval-seconds
```

Please see for more details: Housekeeping Component Remover

ODS - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2025-1-0/release-IPF-2025-1-0-ods.html

ODS - Changes & Fixes

This page covers the changes and fixes provided to ODS IPF Release 2025.1.0

ODS

New

- CAMT055 MDS object supported for the following ODS Inquiry endpoints:
    - `/all/mds-objects/{odsObjectId}`
    - `/all/mds-objects/{mdsObjectId}/history`
    - `/views/details/{unitOfWorkId}`
- Summaries with custom fields are now supported - See the Summary Customisation docs for more information.
    - Summaries can be populated with arbitrary custom fields defined via the existing summary customisation framework
    - Custom fields with a type of String can be marked as searchable, and are searchable via the ODS Inquiry API
- PDS Wrapper added to support PDS versioning in IPF Processing Data
    - Version, metadata and pdsObjectId fields have been included in PdsObjectContainer2
    - Wrapper optionally constructed with version information, metadata and pdsObjectId. This supports manual control of the version information, and is particularly useful in cases where the PDS types cannot be changed easily, e.g. third party dependencies.
    - Annotation added that allows annotation of types with version information and additional metadata. When constructing a wrapper from an annotated type, the wrapper is pre-populated with the annotated values, and wrapper values override any annotated values

Summary Identity Enrichment Housekeeping Task

A new Akka Cluster Singleton can be enabled to execute a recurrent housekeeping task. This task enriches Payment Summary identification fields (debtorAccount, debtorAgentBIC, creditorName, etc…) using the information present on the Payment's related Batch Summary.

This change includes adding a Sparse Index on the new Summary `enrichmentSignal` field. This field is managed by the enrichment logic and is only set on Summaries that require enriching before being removed once those Summaries have been enriched.

> This functionality is disabled by default, it must be enabled via configuring `ods.persistence.housekeeping.summary.identity-enrichment.enabled = true`.

Consult the ODS Ingestion documentation for more information about the enrichment functionality, what configuration is available, and why you may want it enabled/disabled.

Operational Dashboard - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2025-1-0/release-IPF-2025-1-0-operational-dashboard.html

Operational Dashboard - Changes & Fixes

This page covers the changes and fixes provided to Operational Dashboard for IPF Release 2025.1.0

New

Strict Transport Security Header Sent With Each HTTP Response

- The Strict Transport Security header is configured to apply to all requests and all subdomains, ensuring that all traffic is securely served over HTTPS.
- The preload directive is enabled, allowing the domain to be added to the HSTS preload list for enhanced security. (***Note:*** This attribute **has no effect** unless the domain is successfully added to the preload list.)
- The max-age attribute is configurable via the `hsts-max-age` setting.
    - *ipf.business-operations.auth.http.hsts-max-age*
    - please see Determine Processing Entity for more details

> **The `hsts-max-age` Value Must Be Increased**
>
> - By default, max-age is set to **300 seconds (5 minutes)**. The recommended minimum is **31536000 seconds (1 year)**, but it is advisable to increase this value incrementally during deployment.
> - A lower default value provides a grace period for testing. Once a browser enforces HSTS for a domain, it remembers the policy until the max-age expires. Increasing the max-age is simple, but reducing it can be challenging due to browser enforcement.
> - **Best practice** is to start with a low value, test thoroughly, and gradually increase to a long-term secure value. More information can be found here: hstspreload.org/

Functionality to cancel future dated payments

- The functionality to cancel future dated payments has been added to the ODS Payment summary screen. This allows users to cancel payments that have a global status in SCHEDULED or PENDING categories.
- To access this new feature you need to have the following permission: **ROLE_PAYMENT_CANCEL**
- The repair button will only be enabled if the payment global status is in one of the two categories mentioned above. The repair button will not be enabled if the payment global status is in any other category or on the Recall and Bulk summary pages.
- Changes to the global status config and introduction of categories has been explained in the **Changed** section below.

> **NOTE: The cancel functionality has only been added on the frontend, there needs to be a backend implementation added in order for this to work which will be avaliable in the near future.**

Configurable expiry date for JSON Web Token and Cookie

- The duration of how long a JSON Web Token as well as cookie is valid for from the moment of log in can be configured via `ipf.business-operations.auth.jwt.expiry`.
- The user can provide this value given in seconds, minutes, hours and days. The value of the set duration has to fall between 1 minute and 7 days. If this value is misconfigured, the application will not start up.
- The default configuration for this property is 1 hour.

Configurable Signature Algorithm for JSON Web Token

- Signature Algorithm can be configured via setting `ipf.business-operations.auth.jwt.signature-algorithm` along with further required properties for the chosen algorithm to correctly work.
- Supported algorithms are **HS512** and **ES512**.
- `ipf.business-operations.auth.jwt.secret` has to be defined when **HS512** is chosen as signature algorithm.
- `ipf.business-operations.auth.jwt.path-private-key` and `ipf.business-operations.auth.jwt.path-public-key` have to be defined when **ES512** is chosen as signature algorithm.

Granular Permissions Protocol introduced and leveraged by GUI HTM Module

- Granular Permissions Protocol has been created, which is used by the HTM endpoints in the Dashboard.
- The present, role-based permissions remain in use for the endpoints of the rest of the GUI modules.

- `ipf.authorisation.conf` has to be overriden with custom groups, roles and HTM specific contexts to leverage granular authorisation for selected users for access to HTM tasks.
- Default config included in `ipf.authorisation.conf` reuses legacy roles, such as ROLE_HTM_VIEWER, ROLE_HTM_EXECUTE, etc. with permission scopes respectively which replicate authorisation scope as they were used in roles-based protocol.
- The new protocol comes with its own terminology and in order to set up authorisation scopes correctly for HTM GUI module, it is recommended to study the published docs in detail: Granular Authorisation

Changed

Basic Auth Now Disabled by Default

- Basic auth **should not be used in production** and is disabled by default. It is intended only for development and testing purposes. Enable it with:
    - `ipf.business-operations.auth.basic-auth.enabled = true`

Changes to Global Status Config

- The global status config has been changed from an array of strings to an array of objects. Here is an example of what it should look like now:

```
global-statuses = [
   {
       name: "Pending",
       category: "PENDING"
   },
   {
       name: "Accepted",
       category: "ACCEPTED"
   },
   {
       name: "Completed",
       category: "ACCEPTED"
   },
   {
       name: "Rejected",
       category: "REJECTED"
   },
   {
       name: "Manual Action Required",
       category: "MANUAL_ACTION_REQUIRED"
   },
   {
       name: "Scheduled",
       category: "SCHEDULED"
   },
   {
       name: "Cancelled",
       category: "CANCELLED"
   }
 ]
```

> **This is a breaking change** If you have a custom global status config, you will need to update it to match the new format. The name value should be set as whatever you would like the global status to be, and this is the value that will show up on the UI. Each status should be assigned one of 6 categories that has the closest meaning to the global status name. There are only 6 status categories available, and you should not assign a category that does not exist in the following list: **PENDING**, **ACCEPTED**, **REJECTED**, **MANUAL_ACTION_REQUIRED**, **SCHEDULED**, **CANCELLED**. There can be more than one status in each category. Categories have been introduced because they drive certain screen behaviours in the UI, therefore the UI will not work as expected if the category is left null.

Removal of Active User Service

- Following changes from PAY-12897 and PAY-12894, the active user service has been removed.
- Any references to this service should now reference the ngrx store instead as the single source of truth as we no longer decode the JWT locally.

For example, if you used the activeUserService to get the logged in user:

```
this.loggedInUser = activeUserService.getActiveUserInfo();
```

This can be replaced with

```
 this.store.select(selectActiveUser).pipe(
```

```
    map((username: string) =>
        (this.loggedInUser = username)
    ));
```

This is not recommended as we should not be mapping from the selector

- Rather we should handle the loggedInUser as this

```
this.loggedInUser$ = this.store.select(selectActiveUser);
```

Where loggedInUser$ is now an Observable<string | undefined>

- Typically, you don't want to ever reference the logged in user outside of an effect unless it's for display purposes.

Custom Translations Supplied At Release

- Default translations for the IPF Operational Dashboard are supplied at build.
    - Each module has its own scoped translation file, e.g.: `i18n/htm/en.json`
    - In addition, a global translation file now exists for shared or frequently reused terms: `i18n/en.json`
    - At runtime, the global translations are merged into the scoped translations. All keys from the global file are accessible under the scoped namespace. If the same key exists in both the scoped and global files, the **scoped translation takes precedence**.
- Custom translations can be supplied at release time by adding a HOCON file in the following location: `config/bizops/i18n/{scope}/{lang}.conf` (eg. `config/bizops/i18n/htm/en.conf`)
    - This file may include **all or just some** of the scoped translations. It is not mandatory to provide any custom translations - if the file is missing, the application will use the defaults.

> To facilitate this functionality, all translations should be grouped into the following categories:
>
> - navigationItem
> - title
> - heading
> - text
> - columnHeader
> - rowHeader
> - formField
> - tooltip
> - tab
> - button
> - warning
> - error
> - metadataField
>
> Additionally, certain best practices should be followed. Please refer toTranslation Guidelines for more details.

Routing Changes due to cookie change

- The cookie is now httpOnly meaning we no longer have access to it in the FE
- You will need to update the configuration of the providers for your application to handle the changes made on the module

> **This is a breaking change**routerReducer has to be provided for the store module and StoreRouterConnectingModule must also be provided

For example, it should be changed from this:

```
StoreModule.forRoot(
  {},
  {
      runtimeChecks: {
          strictStateImmutability: false,
          strictActionImmutability: false
          }
```

```
    }),
```

To this:

```
StoreModule.forRoot(
  {
      router: routerReducer
  },
  {
      runtimeChecks: {
          strictStateImmutability: false,
          strictActionImmutability: false
          }
  }),
  StoreRouterConnectingModule.forRoot()
```

You will also need to update your routing for the root path so that we are actually checking the user is authenticated and has a valid Processing Entity

```
   {
      path: 'app',
      canActivate: [ProcessingEntityGuard],
      component: AppHomeComponent
  },
```

Migration Steps for IPF-2025.1.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2025-1-0/release-IPF-2025-1-0-migration.html

```
```

Migration Steps for IPF-2025.1.0

Flow Migration

The `ipf-scheduler-floclient-service` module should no longer be used as a dependency, as it has been deprecated in favor of a clearer separation between embedded and remote scheduler implementations.

- To maintain current behavior (embedded scheduler), use the `ipf-scheduler-floclient-service-embedded` module as a dependency
- To use the remote scheduler, depend on the `ipf-scheduler-floclient-service-remote` module instead

Please update your dependencies accordingly to avoid unexpected behavior in future releases.

Migration for Debulker Housekeeping Scheduler

Remove the following configuration parameters if they are defined in your configuration:

```
 ipf.debulker.housekeeping.component.max-age
ipf.debulker.housekeeping.housekeeping-interval-seconds
```

If you wish to override the defaults, define the following configuration parameters:

```
 ipf.debulker.housekeeping.stale-component-remover.component-max-age
ipf.debulker.housekeeping.stale-component-remover.housekeeping-interval
```

> - `ipf.debulker.housekeeping.stale-component-remover.component-max-age`
> - `ipf.debulker.housekeeping.stale-component-remover.housekeeping-interval`
>
> should be specified as a duration — e.g., for 60 minutes, specify `60m`.
>
> For more details on supported formats, see: Duration Format

---

Release Notes for IPF-2025.1.2 - CSM Rulebook Standard Change

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2025-1-2/release-IPF-2025-1-2.html

Release Notes for IPF-2025.1.2 - CSM Rulebook Standard Change

This page details everything required to get going on IPF Release 2025.1.2 made available on Friday 27th June.

This release contains updates to the following Scheme packs developed by Icon:

- **IPF TIPS Scheme Pack** to comply with the annual TIPS scheme upgrade which goes live in the market on **5th October 2025**
- **IPF SIC Scheme Pack** to comply with the SIC IP Rulebook 2025 Updates which are due to go live in the market on the weekend of 22nd November 2025 (exact implementation date and timings to be confirmed)

Please note that this release also includes some improvements to the data model to align it with other Icon standards (*PAY-8240*) and some quality of life bug fixes (*PAY-12080*).

Change Spotlight

Please find all the changes for this release, please see here.

Release Details

In order to build an application using the new changes, change your application pom to use the latest ipf-bom (2025.1.2)

Binaries

The core binaries are available from IPF-Releases.

Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Akka/Alpakka binaries from 2.9.x onwards are no longer on Maven Central and are available from the Akka Library Repository which Icon mirror.

To enable the new mirror, add the following to your settings.xml in the `<repositories>` section:

```
 <repository>
  <id>akka-repository</id>
  <name>Akka library repository</name>
  <url>https://nexus.ipf.iconsolutions.com/repository/akka-repository</url>
</repository>
```

Some MPS dependencies are also mirrored by Icon and can be found by adding the following repository to the same section:

```
 <repository>
  <id>icon-thirdparty</id>
  <name>IPF icon-thirdparty repository</name>
  <url>https://nexus.ipf.iconsolutions.com/repository/icon-thirdparty</url>
</repository>
```

Add the following to your `<servers>` section for each new repository:

```
 <server>
  <id>Repository_name_here</id>
  <username>xxxx</username>
  <password>xxxx</password>
</server>
```

Where `xxxx` is the username and password repectively provided to you by Icon.

Optional Modules - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2025-1-2/release-IPF-2025-1-2-aom.html

Optional Modules - Changes & Fixes

This page covers the optional module changes and fixes provided for both the TIPS and SIC scheme packs to comply with the changes from the respective schemes.

TIPS CSM Scheme Pack

New

- Flag to toggle outbound pacs.008 (debtor and creditor) address validation (*PAY-13673*)
- Flag to toggle inbound pacs.008 (debtor and creditor) address validation (*PAY-13676*)
- Outbound Pacs008 enrichment defaults:
    - Charge bearer field will default to 'SLEV' if null (*PAY-13305*)
    - Settlement method field will default to 'CLRG' if null (*PAY-12636* & *PAY-13305*)
    - EndToEndId will default to 'NOTPROVIDED' if null (*PAY-13924*)

Changed

- Outbound Pacs008 debtor address validator to check for country (*PAY-13477*)
- Inbound Pacs008 debtor address validator to check for country (*PAY-13674*)
- Outbound Pacs008 (debtor and creditor) structured address validator to support hybrid addresses (*PAY-13673*)
- Updated Pacs008 Outbound OrgId validations for UltmtDbtr and Dbtr (Either 'AnyBIC', 'LEI' and/or one occurrence of 'Other' is allowed in OrgId) (*PAY-13481*)
- Inbound Pacs008 (debtor and creditor) structured address validator to support hybrid addresses (*PAY-13676*)
- TIPS Rulebook - limit of 100000 EUR has been removed. Documentation updated to clarify how to toggle/configure amount limit (*PAY-13476*)
- Now **only** checking the `OrgnlMsgNmId` when deciding if this is a `ReceivePaymentSettledRequest` (were previously checking debtor agent BIC not being ours, and `OrgnlMsgNmId`) (*PAY-13886*)
- Updated Pacs008 Inbound OrgId validations for UltmtDbtr and Dbtr (Either 'AnyBIC', 'LEI' and/or one occurrence of 'Other' is allowed in OrgId) (*PAY-13717*)
- Updated Pacs008 Outbound OrgId validations for UltmtDbtr and Dbtr (*Only one occurrence of Other is allowed in CdtTrfTxInf.UltmtDbtr/Dbtr.Id.OrgId*) (*PAY-13481*)
- Updated to the latest XSD Schemas for the October 2025 release:
    - Pacs002 (*PAY-13668*)
    - Pacs008 (*PAY-13474*)
    - Pacs004 (*PAY-13669*)
    - Pacs028 (*PAY-13670*)
- Updated Validations to include return code and system event description for all inbound and outbound validation rules (*PAY-13592*)
- Populate FF01 reason code for outbound validation failures (*PAY-13670*):
    - Pacs.008
        - Interbank Settlement Amount between group header and transaction not matching (GrpHdr+TtlIntrBkSttlmAmt)
        - Either 'AnyBIC', 'LEI' and/or one occurrence of 'Other' is allowed (CdtTrfTxInf+UltmtDbtrId+OrgId)
        - Either 'Date and Place of Birth' or one occurrence of 'Other' is allowed (CdtTrfTxInf+UltmtDbtrId+PrvtId)
    - Pacs.004
        - Number of transactions in group header and underlying transactions do not match (GrpHdr+NbOfTxs)
        - Interbank Returned Settlement Amount between Group Header and transaction do not match (GrpHdr+TtlRtrdIntrBkSttlmAmt)
        - 'Original Group Information' must exist as a child underneath PmtRtr or TxInf, but not both or the message will be

rejected. (OrgnlGrpInf & TxInf+OrgnlGrpInf)

        - Only one instance of PmtRtr.TxInf is permitted (TxInf)

    - Pacs.002

        - GrpSts must exist as a child underneath OrgnlGrpInfAndSts or alternatively TxSts must exist as a child underneath TxInfAndSts, but both should not be present (OrgnlGrpInfAndSts+GrpSts)
        - Status Reason Information can only be populated if GrpSts is Rejected (OrgnlGrpInfAndSts+GrpSts)
        - Original Group Information and Status reason Information must be populated if GrpSts is Rejected (OrgnlGrpInfAndSts+GrpSts)
        - Transaction Information and Status reason Information must be populated if TxSts is Rejected (TxInfAndSts+TxSts)

    - Pacs.028

        - FIToFIPmtStsReq.TxInf.AccptncDtTm is mandatory on a payment status request (TxInf+AccptncDtTm)

> Clients using CSM Reachability should also adjust configured limits in CSM Reachability to align with the TIPS change. For more details, see: Determining Applied Limits

TIPS CSM Simulator

Changed

- Updated tips simulator to be compliant with validation and Rulebook changes (*PAY-14216*)

SIC CSM Scheme Pack

New

- Flag to toggle outbound pacs.008 address validation:
    - Debtor (*PAY-13501*)
    - Creditor (*PAY-13501*)
    - Ultimate Debtor (*PAY-13902*)
    - Ultimate Creditor (*PAY-13902*)
- Flag to toggle inbound pacs.008 address validation:
    - Debtor (*PAY-13503*)
    - Creditor (*PAY-13503*)
    - Ultimate Debtor (*PAY-13903*)
    - Ultimate Creditor (*PAY-13903*)
- New validation rule on outbound ultimate creditor and ultimate debtor addresses. (*PAY-13902*)
- New validation rule on inbound ultimate creditor and ultimate debtor addresses. (*PAY-13903*)
- Outbound Pacs008 enrichment defaults:
    - Local instrument proprietary field (pmtTpInf.lclInstrm.prtry) will default to 'IPCPMT' if empty (*PAY-13792*)
    - EndToEndId will default to 'NOTPROVIDED' if null (*PAY-13924*)

Changed

- Outbound Pacs008 (debtor and creditor) structured address validator to support hybrid addresses (*PAY-13501*)
- Inbound Pacs008 (debtor and creditor) structured address validator to support hybrid addresses (*PAY-13503*)
- Updated SIC Validations to include return code and system event description for all inbound and outbound validation rules (*PAY-13592*)

Fixed

- Reviewed/corrected SIC Validations documentation for SIC (*PAY-14208*)

SIC CSM Simulator

Changed

- Updated sic simulator to be compliant with validation and Rulebook changes (*PAY-14220*)

Release Notes for IPF-2024.4.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-4-0/release-IPF-2024-4-0.html

Release Notes for IPF-2024.4.0

This page details everything required to get going on IPF Release 2024.4.0 made available on **21st February 2025**.

Change Spotlight

> Spring has been upgraded to **3.4.2**. The Spring Release notes can be found here.

- IPF serialization using SerializationHelper is now configured by Akka Jackson Serialization, allowing further customisation via Hocon. Dependency to `com.iconsolutions.ipf.core.shared:shared-serialization` requires additional transitive dependencies to Akka (*PAY_3828*).
- Changed default configuration for `propagate-transport-errors` to `false` for IPF Processing Data System Event Exporter, Direct Data Exporter and Message Logger (*PAY-11612*).
- Default date format in the GUI is now configurable (*PAY-9952*).
- New controls in place to restrict operators from approving their own modifications to the system in the GUI (*PAY-12579*).
- Added acmt.023.001.04 & acmt.024.001.04 message types to the IPF ISO model.
- Added ability to perform single and multiple duplicate checks in duplicate check business function.
- `TransactionCacheEntryType` can now be set from a mapping function or using the meta data tag in duplicate check business function.
- Added ability to control when job rehydration occurs in persistent scheduler. If `ipf.persistent.scheduler.automatically-rehydrate-jobs` set to `false`, then rehydration only occurs explicitly via a command sent to the `JobRehydratorActor`.
- Updated liveness health check docs for use in active/passive clusters (*PAY-12551*).

Fix Spotlight

- Test-FW:
    - Fixed IS_SET to correctly assert values that are set or not. Note that you need to assert on fields, and currently we do not support asserting on complex objects.
    - Fixed story reporter to show executed test steps as they are progressed in the IDE.
- HTTP connector transports now self-recover from unhandled exceptions thrown during the request-response cycle, meaning that the restart of the application instance is no longer required to resume sending requests (*PAY-11809*).
- `MessageSent` events are no longer erroneously raised when sending a message via a send connector fails (*PAY-12010*).
- Fixed `dot-compiler` plugin build step failures caused by very large MPS payment flows (*PAY-12756*).
- Fixed MPS being unable to visualize flows for very large MPS payment flows (*PAY-12757*).
- Fixed cases of duplicate notifications for debulker file poller when using multiple instances (*PAY-12145*).
- FilePollerScheduler creating duplicate scheduled jobs on startup (*PAY-12356*).

Non-Breaking Changes

- Akka Discovery MongoDB plugin: Configuration property akka.discovery.akka-mongodb.enabled is now persisted on receiving a POST /discovery/cluster-status request. On application startup, the persisted value is used if it exists. Otherwise, the value from the configuration file is used. No additional configuration is required.
- `SerializationHelper` is now configured by Akka Jackson, which means its behaviour can be overridden with Hocon. See the Serialization documentation for more information.
- Retries are now configured by default for spring reactive repositories in projects with the `ipf-common-starter-mongo` dependency (*PAY-12783*). For more details see Mongo DB Starter

Breaking Changes

- Iso 20022 model package changed for class *com.iconsolutions.iso20022.message.components.technical.remittance_amount3.RemittanceAmount3* to *com.iconsolutions.iso20022.message.components.document.remittance_amount3.RemittanceAmount3*
- `SerializationHelper.objectMapper()` now returns a singleton instance of `ObjectMapper`, which in itself is not a breaking change, but may result in unexpected behaviour if the instance is configured/mutated. In those cases, use `ObjectMapperFactory.createObjectMapper()` instead.

Release Details

Binaries

The core binaries are available from IPF-Releases.

Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Akka/Alpakka binaries from 2.9.x onwards are no longer on Maven Central and are available from the Akka Library Repository which Icon mirror.

Scaffolder

The latest version of the Scaffoler is **1.0.6**

Developer App

The latest version of the Developer App is **2.3.31**

2024.4 Jar/Pom list

The Jars and Poms for 2024.4.0 and the associated versions are listed here: 2024-4-artifacts.xlsx

---

Optional Modules - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-4-0/release-IPF-2024-4-0-aom.html

Optional Modules - Changes & Fixes

This page covers the optional module changes and fixes provided in release IPF-2024.4.0.

Scheme Packs

SIC CSM

Fixed

- Validate SCOR reference when Issr = ISO to adher to ISO 11649 requirements, for outbound transactions (*PAY-11988*)

SEPA CT CSM

Fixed

- For SEPA Credit Transfers, the Transaction Reason Proprietary Code will be set to T00 instead of B00 when the Transaction Status is one of the following:
    - ACTC: After a successful IPF validation.
    - ACSP: Upon receipt of a CVF that results in either a Group Status (GrpSts) of ACCP or PART.
- Additionally, the Transaction Reason Proprietary Code is set to XT00 for SEPA Credit Transfers when Pacs.002 is generated during End of Day (EOD) processing or when a RSF is received with a GrpSts of either ACCP or PART.
- Serbia has been removed from the list of EEA countries

Removed

- SepaXMLUtils has been superceded by XMLUtils which is available by adding the following dependency

```
<dependency>
   <groupId>com.iconsolutions.ipf.core.shared</groupId>
   <artifactId>shared-xml</artifactId>
</dependency>
```

TIPS CSM

Fixed

- Pacs008 amount validation incorrectly comparing values with trailing 0's (*PAY-12405*)
- Postal address sub-field validation incorrectly failing for unstructured addresses where country not provided (*PAY-12430*)
- Ignore sending Pacs.002 confirmation messages from the TIPS simulator when an inbound Pacs.008 is rejected due to failure. (*PAY-11318*)

Verification of Payee

New

- Added Verification Of Payee Requester API
- Added Account Management API

> In this release the above APIs are provided as a reference. The implementation of these are ongoing and will be delivered in a future release.

Identity Resolution

New

- New module called `comparison-client-direct` added to allow embedding identity resolution into a service

Changed

- We no longer send requests to Netowl if the comparison only contains special characters. Instead, a score of 0.0 will be returned for these comparisons. New config value of *identity-resolution.comparison.netowl.exclusion.special-characters*, which takes a list of

characters, has been added should you want to add or remove characters from the list. (*PAY-11884*) Default list includes some Basic Latin special characters:

```
identity-resolution.comparison.netowl.exclusion.special-characters =
[
"/", "\\", "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", "\"", "{", "}", "_", "[", "]" ,"|", "?", "<", ">", ",", "." " "
]
```

- The Identity Resolution API now supports null and empty values. It returns a positive response (match = true) when both the actual and comparison values are null or empty, or when the match threshold is set to 0.0.

Metrics Processor

IPF Label Support

IPF labels belonging to an event or a state are supported and when present are used to determine the critical path duration and waiting duration metrics, and when labelling payment metrics with HTM="Yes".

The existing configuration-based approach continues to work until labels are produced by flows, which means the IPF label support change is generally non-breaking. However, take note of the following to avoid confusion:

- When there is existing configuration that defines waiting states for several different flows for a given payment, and one of those flows is updated in MPS with labels, the configuration for the other flows is ignored and only the labels are used to determine the waiting duration. This will result in a smaller waiting duration. **When updating flows to enable labels, all possible flows must be updated.**
- The existing configuration for the critical path duration is based on the global status, but labels can only be applied to states and events. The effect is the same provided that the labelled flow state or event corresponds to the previously configured global state.
- It is now possible for multiple critical paths to be defined, within multiple different flows, e.g. states `A → B → C → D → E → F` where `A → B` and `D → E` are labelled with `CRITICAL_PATH_START` AND `CRITICAL_PATH_END` respectively. The sum of the critical paths is used when producing the final metric.
- For metrics labelled with `{ HTM="Yes"}`, Both labelled events/states, and the existing configuration is used.

> In all cases the existing configuration outlined above is considered deprecated, and will be removed in a future release.

Client Channel Label Support

The `clientChannel` label has been added to emitted payment metrics to represent the client payment initiation channel used for a payment. The value can be sourced from any client-specific PDS object. An example client channel mapping configuration is:

```
ipf.business-metrics-processor.payment-metrics.labels {
 client-channel {
     pdsType = ClientSpecificType
     path = "instrMsg.cstmCrdTrfInitn.chanl"
  }
}
```

For more information, see the client channel metrics processor documentation.

Working Day Service

Fix spotlight

- OpenAPI related specification fixes, including:
    - `AnnualDate` now being correctly rendered
    - Using `OffsetTime` timestamp format in examples
    - Specification now being rendered correctly in direct starter project

Human Task Manager

New

- Alpha Feature: search query parameter of `metaDataTags` added to API. Performance checks are scheduled to be carried out.

CSM Reachability - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-4-0/release-IPF-2024-4-0-csm-reach.html

CSM Reachability - Changes & Fixes

This page covers the CSM Reachability V2 improvements, changes and fixes provided in release IPF-2024.4.0.

CSM Reachability and Industry Data Ingestion

CSM Reachability API

New

- On DPS Setting management API for Agent Settlement Settings, added new attribute for Limit Direction (Enums: Inbound, Outbound). Applicable for limit types supported (Agent Limits, Custom Participant Limits, Custom Country Limits)

CSM Reachability

Changed

- LimitDirection attribute taken into account when processing validate CSM Reachability and Select CSM Agent requests where only outbound limits apply where configured.
- On the domain API, /v2/settlement-agents, the limit values returned on the response match with the transfer direction on the request, where corresponding limits are configured (i.e. outbound limits are returned when the input transfer direction is "Outbound" and inbound limits are returned when input transfer direction is "Inbound")
- Add additional check if the agent limit direction is `OUTBOUND in IBAN reachability reachableByCurrencies check.

Data Ingestion

Changed

Below migration job is available for configuring the limit direction as "Outbound" for all the existing limits that are configured by client implementations.(all existing limits are by default outbound). This migration job is optional and the existing limits will be treated as outbound even if the migration job is not run.

Migration job details available here.

Bulker & Debulker - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-4-0/release-IPF-2024-4-0-Debulker.html

Bulker & Debulker - Changes & Fixes

This page covers the Bulker & Debulker module changes and fixes provided in release IPF-2024.4.0.

Debulker

Changed

- Now when a failure occurs during debulking, processing transitions to a `Rejected` state and a system event is emitted.

Fixed

- Housekeeping job scheduling has been delegated to an Akka cluster singleton, ensuring duplicate housekeeping jobs are no longer created on each application start/restart (*PAY-12395*)

File Poller - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-4-0/release-IPF-2024-4-0-file-poller.html

File Poller - Changes & Fixes

This page covers the File Poller module changes and fixes provided in release IPF-2024.4.0.

New

Changed

Fixed

- Duplicate file poller's being created on each application start/restart (*PAY-12356*)
- File poller now works correctly in a clustered environment. (*PAY-12145*)

Removed

ODS & GUI - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-4-0/release-IPF-2024-4-0-gui-ods.html

ODS & GUI - Changes & Fixes

This page covers the Operation Data Store (ODS) and GUI changes and fixes provided in release IPF-2024.4.0.

ODS

New

- Support added to return `NON_ISO` MDS objects for the following ODS Inquiry endpoints:
    - `/api/v2/all/mds-objects`
    - `/api/v2/all/mds-objects/{odsObjectId}`
    - `/api/v2/all/mds-objects/{mdsObjectId}/history`
    - ***NOTE:*** The equivalent ODS Inquiry V1 endpoints also support returning `NON_ISO` MDS Objects.
- `processObjectId` search parameter added to the following ODS Inquiry endpoints:
    - `/api/v2/all/process-objects`
    - `/api/v2/catalogue/process-objects/message-logs`
    - `/api/v2/catalogue/process-objects/process-flow-definitions`
    - `/api/v2/catalogue/process-objects/system-events`
    - `/api/v2/catalogue/process-objects/process-flow-events`
    - ***NOTE:*** The equivalent ODS Inquiry V1 endpoints also support searching by `processObjectId`.
- ODS Ingestion core summary mapping added for `pacs.004` mds objects to the Summary `instructionReceivedAt` field. See core summary mappings.

GUI

New

- **Selection Order to be Configurable Dynamically by Clients in the IPF GUI:**
    - Agent Selection Settings can be viewed, edited, deleted and approved from the Agent Selection Settings page.
    - Agent Selection Options can be reordered by dragging and dropping or by changing the number in the order box.
- **Search Process Object Catalogue (System Events and Message Logs):**
    - Users can now search for and view all system events and message logs stored within the library.
        - For more information, see the documentation on message logs and system events.
    - On the view page for a given process event, related objects are linked and can be navigated to by clicking on the ID.
- **Improve Role Based Access to IPF Components:**
    - Metadata tags have been added to HTM tasks.
    - The metadata tags can be dynamically attached to a HTM Request based on data from the flow.
    - A search query parameter of `metaDataTags` has been added to the API.
- **Core Project Defects:**
    - Block users from approving records they have created, edited or deleted in processing settings, this is configurable (default set to true) and can be switched off.
        - Auditing has been added for the deletion of a dynamic processing setting.
- **Configuration Changes:**
    - HTTP client host and port transport configuration, requiring of approval for deletion, creation and updating, call time-out and resiliency-settings added for CS Agent Selection Settings:
        - *ipf.business-operations.processing-settings.cs-agent-selection-settings.http.client.host*

        - *ipf.business-operations.processing-settings.cs-agent-selection-settings.http.client.port*
        - *ipf.business-operations.processing-settings.cs-agent-selection-settings.http.client.deletion-requires-approval*
        - *ipf.business-operations.processing-settings.cs-agent-selection-settings.http.client.creation-requires-approval*
        - *ipf.business-operations.processing-settings.cs-agent-selection-settings.http.client.update-requires-approval*
        - *ipf.business-operations.processing-settings.cs-agent-selection-settings.http.client.call-timeout*
        - *ipf.business-operations.processing-settings.cs-agent-selection-settings.http.client.resiliency-settings*
    - Added new permissions roles for reading, creating, updating, approving and deleting CS Agent Selection Settings:
        - *ROLE_DPS_SO_R*
        - *ROLE_DPS_SO_C*
        - *ROLE_DPS_SO_U*
        - *ROLE_DPS_SO_A*
        - *ROLE_DPS_SO_D*
    - Configuration for system event names to be used in dropdowns on ODS Search in the UI:
        - *ipf.business-operations.payment-search.system-event-names*
    - Configuration for whether the creation, edit or deletion of a processing setting has to be authorized by another user. This is a boolean and default is true.
        - *ipf.business-operations.processing-settings.requires-other-approval*

Changed

- **Core Project Defects:**
    - Critical user information (processing entity and username) is taken from the headers or JWT token (rather than the payload) when used to create a record, this is for security purposes to ensure data can not be falsely created, edited, approved, deleted or accessed.
    - The executionResultData property is now being mapped correctly when bulk executing HTM tasks.
    - Payment Search now returns correct results when searching by date.
    - User UI can now define how many tasks are seen in HTM search screen.
    - In Payment Search, search by originalInterbankSettlementDate results are accurate.
    - The ODS export button is now disabled when there are no results to export.
    - Searching for an audit record by ID now works.
    - ODS Summary Page alignment has been corrected.
    - In ODS Summary, mouse over now correctly shows related text.
- **GUI Improvements:**
    - In Agent Settings, a blank option is now included for Identifier Type to enable the user to deselect an Identifier Type.
    - Descriptions are now displayed against user roles within permissions module.
    - Thousands separators present in all amount fields.
    - Breadcrumb font sizing is now the same for both link and title.
    - Pagination added to HTM search results.
    - Approval button no longer shows in action column when bank filtering rule has Active status.
    - Fixed issue with delay when setting up forms.
    - Processing Settings dialog alignment has been corrected.
    - Agent Settlement Settings form has been expanded to include the following additional fields:
        - Account Servicing Agent: Agent Country

        - Agent Limits: Limit Direction
        - Custom Participant Limits: Limit Direction
        - Country Limits: Limit Value, Limit Type, Country, Limit Direction
    - InnerHtml directive has been removed/content is checked before render.
    - Gui-service exception responses are now uniform.
    - Default date format is now configurable.
    - Updated jjwt dependency to the latest version (0.12.6).

> This is a **BREAKING CHANGE** as the JWT token has been changed due to this update. Previously the token looked like this:
>
> ```
> {
>  "alg": "HS512"
> }
> {
>  "sub": {
>  "password": null,
>  "username": "username",
>  "authorities": [],
>  "accountNonExpired": true,
>  "accountNonLocked": true,
>  "credentialsNonExpired": true,
>  "enabled": true
>  },
>  "iat": 1736766003028,
>  "roles": {
>  "BANK_ENTITY_1": [
>  "ROLE_1",
>  "ROLE_2"
>  ]
>  }
> }
> ```
>
> Now the token has been made smaller, 'sub' has been changed to align to a more standard JWT token layout:
>
> ```
> {
>  "alg": "HS512"
> }
> {
>  "sub": "username",
>  "iat": 1737129789,
>  "roles": {
>  "BANK_ENTITY_1": [
>  "ROLE_1",
>  "ROLE_2"
>  ]
>  }
> }
> ```
>
> Please follow migration steps below to update.

- **ODS Payment Search module name change**

  > Both front-end and back-end have had a name change of the payment-search modules to be ods in order to match with its actual use. The ods-payment-search module and npm package have been renamed to ods. This is a breaking change, please follow migration steps to update.

    - In ops-gui-service-ng the new modules are ops-gui-service-ng-ods-parent, ops-gui-service-ng-ods-api, ops-gui-service-ng-ods.

- **Angular 18**

  > As of 2024.4 the Operational Dashboard and all related modules have been updated to Angular 18, please follow migration steps below to update.

- **ODS Export date format is now configurable**

  > This is a breaking change as a new parameter has been added to the request, please follow migration steps below to update.

- **Configuration Changes:**
    - Made number of HTM tasks displayed on the search page configurable.

        - *ipf.business-operations.human-task-manager.htm-page-results*
    - Date formats are now configurable by providing the following injection tokens in the `app.module.ts`:
        - *IPF_DATE_DISPLAY_FORMAT*
        - *IPF_DATE_TIME_DISPLAY_FORMAT*
        - *IPF_DATE_TIME_INPUT_FORMAT*
        - *IPF_DATE_INPUT_FORMAT*

Removed

N/A

Migration Steps

- Bump `ipf-release-ops-gui-bom` to `2024.3.1.140` in pom.xml
    - As part of this change the Subject class (com.iconsolutions.ipf.gui.core.auth.util.Subject) was replaced with the Spring User class (org.springframework.security.core.userdetails.User) for conformity across auth methods.
    - As the auth mechanism has been updated slightly, you may need to clear cache or close existing sessions if you are working during this upgrade.
- Bump `icon-dependency-management-java17` to `2.2.0` in pom.xml
- Update your Angular Dashboard to Angular 17 angular.dev/update-guide?v=16.0-17.0&l=1
- Update your ngRx to be 17 ngrx.io/guide/migration/v17
- Update your Angular Dashboard to Angular 18 angular.dev/update-guide?v=17.0-18.0&l=1
- Update your ngRx to be 18 ngrx.io/guide/migration/v18
- Update your Angular Dashboard to change the ODS dependency:

<artifactId>ops-gui-service-ng-payment-search</artifactId>

to

<artifactId>ops-gui-service-ng-ods</artifactId>

- Update your Angular Dashbaord's app-routing.module.ts file and change the route to be:

```
  {
    path: 'ods',
    data: { roles: ['ROLE_PAYMENT'] },
    canActivate: [RoleGuard],
    loadChildren: () => import('@iconsolutions/ods').then((m) => m.OdsModule)
  }
```

- Update @iconsolutions/* npm packages as detailed below:

@iconsolutions/audit

- Bump npm module to `"@iconsolutions/audit": "^18.0.3"` in package.json
- Refactor `i18n/audit/*.json` to reflect new json structure removing configuration details from translation sheet.

"auditColumns": [{ "name": "time", "label": "Created", "type": "moment" }, …]

"auditColumn": { "time": "Created" …

@iconsolutions/cluster-health

- Bump npm module to `"@iconsolutions/cluster-health": "^18.0.0"` in package.json

@iconsolutions/common

- Bump npm module to `"@iconsolutions/common": "^18.2.3"` in package.json
- See docs around new injection tokens for date formats

@iconsolutions/ods

- Change `"@iconsolutions/ods-payment-search"` npm module to `"@iconsolutions/ods": "^18.1.4"` in package.json
- Get latest translation file from transloco scoped libs runner
- Change any payment search api references from `./api/paymentsearch` to `./api/ods`

@iconsolutions/htm

- Bump npm module to `"@iconsolutions/htm": "^18.0.1"` in package.json

@iconsolutions/metrics

- Bump npm module to `"@iconsolutions/metrics": "^18.0.0"` in package.json

@iconsolutions/permissions

- Bump npm module to `"@iconsolutions/permissions": "^18.0.1"` in package.json
- Get latest translation file from transloco scoped libs runner

@iconsolutions/processing-settings

- Bump npm module to `"@iconsolutions/processing-settings": "^18.0.11"` in package.json

@iconsolutions/version-info

- Bump npm module to `"@iconsolutions/version-info": "^18.0.1"` in package.json

Migration Steps for IPF-2024.4.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-4-0/release-IPF-2024-4-0-migration.html

Migration Steps for IPF-2024.4.0

Migration Steps for CSM Reachability

Limit Directions on Agent Settlement Settings

> This migration script is optional, however, it is highly recommended to run the following to ensure compatibility with future changes. This is because limit direction has been added as an option on Agent Settlement settings.

If the script is run, the `limitDirection` field will be assigned a default value of `OUTBOUND`. If the script is not run, `limitDirection` will remain unset in the database, but will still be treated as `OUTBOUND` in the code.

```
db.getCollection("settings-agent-settlement-settings")
    .updateMany(
        { "payload.agentLimits.limitValue": {$exists: true} },
        { $set: { "payload.agentLimits.$[elem].limitDirection" : "OUTBOUND"}},
        { arrayFilters: [ { "elem.limitDirection": { $exists: false} } ] }
    )

db.getCollection("settings-agent-settlement-settings")
    .updateMany(
        { "payload.customParticipantLimits.limitValue": {$exists: true} },
        { $set: { "payload.customParticipantLimits.$[elem].limitDirection" : "OUTBOUND"}},
        { arrayFilters: [ { "elem.limitDirection": { $exists: false} } ] }
    )

db.getCollection("settings-agent-settlement-settings")
    .updateMany(
        { "payload.participantCountryLimits.limitValue": {$exists: true} },
        { $set: { "payload.participantCountryLimits.$[elem].limitDirection" : "OUTBOUND"}},
        { arrayFilters: [ { "elem.limitDirection": { $exists: false} } ] }
    )
```

Migration Steps for the Metrics Processor

Elements of the IPF Metrics Processor configuration have been decoupled from release-specific MPS flow versions and names.

Previously, when payment flows get a version bump or names of events or (global) states change, the configuration of the metrics processor would need to be updated in line with these changes, otherwise payments metrics would not be labeled correctly in Prometheus.

Three of Metrics Processor configurations that were based on flow versions and state names have been removed, with the specification of these being moved to the MPS Flow Definition.

These three config values require:

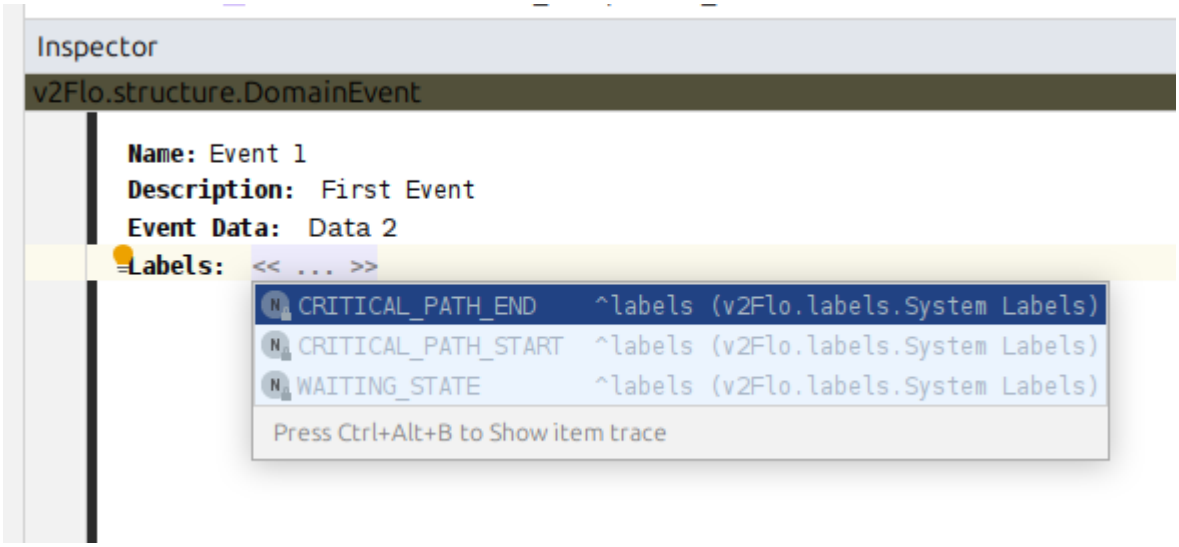
- `critical-path-duration` - requires the configuration of global state names per payment type
- `waiting-duration.waiting-states` - used indicate where the payment flows had to wait for external responses
- `tagged-with-htm-on-any-of` - used to indicate that a payment went into HTM

Migration is currently optional, with the Metrics Processor still supporting the above Hocon configuration keys. However, it is recommended to update your MPS Payment Flows with the appropriate labels to indicate critical path and waiting states, then remove the existing Metrics Processor configuration.

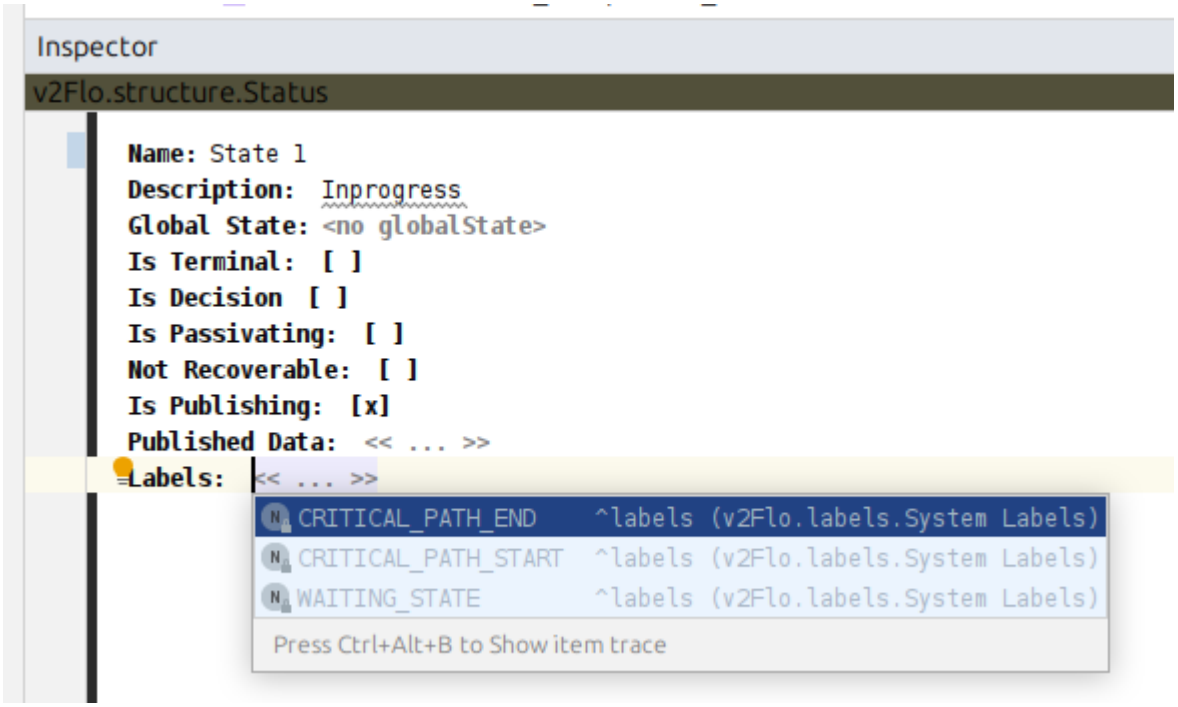
Migration requires using MPS Flow Labels as a replacement for the HOCON configuration, see the Flow Orchestration documentation for more information about Labels.

Labels can be added to both State Definitions and Event Definitions using the MPS Inspector window like so:

- Inspector Window for Event Definitions



- Inspector Window for State Definitions



Example Migration

The following steps will reference flow names and states from the below example Metrics Processor HOCON configuration as a guide to migrating your MPS Flow definitions. These flow names and states are purely an example, your existing implementation will be different.

Example Configuration

```
ipf.business-metrics-processor.payment-metrics {
 htm {
   events = ["Manual Intervention", "Registered Task" ]
 }

 payment-duration {
   critical-path {
     critical-path-states-by-payment-type = [
       {
         payment-type = "InstantPayment"
         start-state = "Validating"
         end-state = "Instructing"
       }
     ]
   }

   waiting {
     waiting-states-by-flow = [
       {
```

```
          flow-name = "DebtorCT"
          states = ["Checking Bank System A", "Checking Bank System B", "Waiting for Response"]
        }
      ]
    }
  }
}
```

Critical Path Duration

To migrate the critical path configuration for the above example, MPS flows that comprise the `InstantPayment` payment type need to be updated.
Within these MPS Flows, the flow states that line up with the global states `Validating`, and `Instructing` should be updated with the labels `CRITICAL_PATH_START` and `CRITICAL_PATH_END` respectively.

Once this change has propagated to downstream applications, it is safe to delete the Metrics Processor `critical-path` configuration block.

Waiting Duration

To migrate the waiting configuration for the above example, the `DebtorCT` MPS Flow needs to be updated.
The `WAITING_STATE` label should be added to the `Checking Bank System A`, `Checking Bank System B` and `Waiting for Response` states within the `DebtorCT` MPS Flow.

Once this change has propagated to downstream applications, it is safe to delete the Metrics Processor `waiting` configuration block.

HTM metric label

The `HTM` label is only available when using the Human Task Manager Flo Client to integrate with the Human Task Manager application.
To migrate the HTM configuration for the above example, the `Manual Intervention` and `Registered Task` events within all MPS Flows should be updated with the `HTM` label.

Once this change has propagated to downstream applications, it is safe to delete the Metrics Processor `htm` configuration block.

Migration Steps for File Poller

File poller has been updated to cleanup and recreate file poller scheduler jobs on application startup. As part of this change, the values for `jobRequestor` and `jobSpecificationId` fields have been updated. New and previous values are listed below.

Field	Previous	New
`jobRequestor`	`filePoller`	`ipfFilePoller`
`jobSpecificationId`	Prefixed with `file-scheduler.`	Prefixed with `file-poller-scheduler-` and the value of configuration item `ipf.file-poller.application-id`. For example the default prefix would be `` `file-poller-scheduler-default-id ``

Since the cleanup logic implemented is based on the new values above, existing jobs in the `jobSpecification` and `jobExecutionStatus` collections in mongo will need to be manually removed from these collections.

Migration Steps for Egress Processing Data

The default configuration for IPF Processing Data plugins for System Event Exporter, Direct Data Exporter and Message Logger for `propagate-transport-errors` has been changed to `false`.

If you wish to continue to propagate transport errors you must update your configuration to set `propagate-transport-errors` to `true` for each buffered exporter.

Example Configuration

```
ipf.processing-data.egress{
 system-events {
   buffered-exporter {
     propagate-transport-errors = true
   }
 }
 message-logger {
   buffered-exporter {
     propagate-transport-errors = true
   }
 }
```

```
  direct-exporter {
    buffered-exporter {
      propagate-transport-errors = true
    }
  }
}
```

Migration Steps for SerializationHelper

All calls to `SerializationHelper.objectMapper()` where the `ObjectMapper` instance it returns is customised, e.g. by calling `setSerializationInclusion()` , must be replaced with `ObjectMapperFactory.createObjectMapper()` which returns a new instance.

See Serialization documentation for more information

Release Notes for IPF-2024.3.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-3-0/release-IPF-2024-3-0.html

Release Notes for IPF-2024.3.0

This page details everything required to get going on IPF Release 2024.3.0 made available on 20th November 2024.

Change Spotlight

> You must choose whether to start exporting IPF Processing Data in the V2 schema (the default if you do nothing), or in the existing V1 schema. To pin/fix the data model version, use `ipf.processing-data.egress.schema-version = 1` to stay with the current version, or `ipf.processing-data.egress.schema-version = 2`.
>
> See the migration guide for Migration Steps for IPF Processing Data Version 2.

> - Archetype has now been replaced by **ipf-project-scaffolder**
>
> The latest version of the scaffolder is **1.0.5**
>
> Instructions on how to use the scaffolder to create new projects are here

Fix Spotlight

- More descriptive error handling in Identity Resolution (*PAY-9675*)
- Circuit Breakering and Retry configuration is now more flexible and intuitive to configure (*PAY-4362*)

Non Breaking Changes

- The IPF Processing Data ingress receive connector supports handling a batch of records when using Kafka. Handling a batch of IPF Processing Data envelopes is opt-in and requires a code-change supplying a batch handler. The existing non-batched handler continues to be supported.
- `reference.conf` added to the configuration hierarchy for IPF modules (where configuration is local to the module and not for a dependency)

Breaking Changes

- As part of the continued drive for better re-usability across DSL components, response and reason codes are now only generated once for the model that they are defined within. This means that the packaging of these classes may change, particularly if using the core 'AcceptOrReject' or 'IsoReasonCodes'. See the migration guide for further details.
- IPF produces IPF Processing Data in the V2 schema by default. To remain on V1 set `ipf.processing-data.egress.schema-version = 1`. See the migration guide for further details.
- As part of the IPF test framework update, in class `com.iconsolutions.ipf.core.test.kafka.KafkaTestTransporter`, `withProducerConfig` and `withConsumerConfig` have been removed from the builder. `withPropertiesPath` should be used to build this class.
- Validation interfaces moved into separate API. Previously in Clear and Settle API - customers using the validation api, will also need to add the `validation-api` jar
- Extension Point API and client port moved into its own API project (previously was contained in SEPA CT API). SEPA CT API, now depends on the Extension Point API project
    - Specifications for both V1 and V2 can be found here.
- Notification-service interface `AdditionalPaymentObjectHandler` updated to decouple from IPF processing data model. See notification-service changes for migration details
- IPF Archiver produces archive bundles in the V2 IPF Processing Data schema by default. If you want to continue to use the V1 schema, it must be enabled with `ipf.archiver.bundle.schema-version = 1`.
- To ensure BigDecimal datatypes in business data that have values with trailing zeros (e.g. `15.00`) no longer lose precision during payment processing or when this data is exported to ODS, the jackson deserialization feature USE_BIG_DECIMAL_FOR_FLOATS has been **enabled by default** in:
    - the akka serialization config in `ipf-common-starter-core` via the parameter:
        - `akka.serialization.jackson.deserialization-features.USE_BIG_DECIMAL_FOR_FLOATS=on`
    - the `SerializationHelper` utility class via the ObjectMapper property:
        - `DeserializationFeature.USE_BIG_DECIMAL_FOR_FLOATS`
- If you wish to override this new default behaviour:

    - add the following config parameter to your application config file:
        - `akka.serialization.jackson.deserialization-features.USE_BIG_DECIMAL_FOR_FLOATS=off`
    - Create a custom object mapper implementation using the `SerializationHelper.objectMapper()` method, and disable the `DeserializationFeature.USE_BIG_DECIMAL_FOR_FLOATS` property:
        - `SerializationHelper.objectMapper().disable(DeserializationFeature.USE_BIG_DECIMAL_FOR_FLOATS)`
- `withResiliencySettings(ResiliencySettings resiliencySettings)` has been deprecated and has been replaced with `Function<ResiliencySettings, ResiliencySettings> resiliencySettingsCustomiser`

Release Details

Binaries

The core binaries are available from IPF-Releases.

Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Akka/Alpakka binaries from 2.9.x onwards are no longer on Maven Central and are available from the Akka Library Repository which Icon mirror.

To enable the new mirror, add the following to your settings.xml in the `<repositories>` section:

```
 <repository>
  <id>akka-repository</id>
  <name>Akka library repository</name>
  <url>https://nexus.ipf.iconsolutions.com/repository/akka-repository</url>
</repository>
```

Some MPS dependencies are also mirrored by Icon and can be found by adding the following repository to the same section:

```
 <repository>
  <id>icon-thirdparty</id>
  <name>IPF icon-thirdparty repository</name>
  <url>https://nexus.ipf.iconsolutions.com/repository/icon-thirdparty</url>
</repository>
```

Add the following to your `<servers>` section for each new repository:

```
 <server>
  <id>Repository_name_here</id>
  <username>xxxx</username>
  <password>xxxx</password>
</server>
```

Where `xxxx` is the username and password repectively provided to you by Icon.

Developer App

The latest version of the Developer App is **2.3.23**

2024.3.0 Jar/Pom list

The Jars and Poms for 2024.3.0 and the associated versions are listed here: 2024-3-artifacts.xlsx

---

Core - Improvements, Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-3-0/release-IPF-2024-3-0-core.html

Core - Improvements, Changes & Fixes

This page covers core improvements, bug fixes, changes and fixes provided in release IPF-2024.3.0.

Core improvements, bug fixes & changes

New

- **IPF Processing Data**
- Version 2 of the IPF Processing Data model has been introduced. Specifications for both the V2 and V1 data models can be found here.
    - The IPF Processing Data Egress plugins are set, by default, to export the V2 data model version. Only one version will be exported for any given application that uses the egress plugins. However, you can configure the plugins to use the V1 data model if necessary using `ipf.processing-data.egress.schema-version = 1`.
    - IPF Processing Data Ingress plugins can consume both the V2 and V1 data model at the same time. The implementing application should provide a handler for the new data model. See the migration guide for further details

Changed

- **Connector**:
    - when an ActorSystem is not passed to a connector, an error message is logged instead of a warning message
    - all Send/Receive connectors' builders use debug logging level when dependency is not provide and default one is used
    - all Send/Receive connectors' builders use debug logging level when configuration parameter is not provide and default one is used
    - all Send/Receive connectors' builders use the same logging message formatter

Fixed

- **dot-compiler** maven plugin - updated dependency version to address the issue with flow builds on Windows (j2v8 engine fails to parse a file); the new version is absorbed in **connector**, **flo-lang** (used in tiles), and **ipf-release-management**

Notification Service

New

Changed

- The `AdditionalPaymentObjectHandler` has been updated and no longer takes a parameter of type `com.iconsolutions.ipf.processingdata.mds.MdsObjectContainer`.There are two changes of note:
    - The interface has been renamed to `AdditionalMdsObjectContainer`
    - The first parameter of the `handle` method has been updated to accept a parameter of type `com.iconsolutions.ipf.product.notification.api.model.MdsObjectWrapper`
        - Example of building a `MdsObjectWrapper` from a `MdsObjectContainer`:

```
MdsObjectWrapper<?> toMdsObjectWrapper(final MdsObjectContainer<?> container) {
    return new MdsObjectWrapper<>(container.getObjectType(), container.getObject());
}
```

- Previous 2024.2 docs → Current docs

> Message Logs produced by the Notification Service app utilise the V2 IPF Processing Data Model. To remain on the V1 model set `ipf.processing-data.egress.schema-version = 1`.

Human Task Manager

New

- HTM will now publish all the new domain events to IPF Processing Data and to ODS by default.

    - To disable events being sent, add `ipf.journal.exporter.type = none` to your `application.conf`
    - In case you wish to export all of HTM domain events from the beginning of history, you can do so by adding `event-processor.start-stream-from = EARLIEST` to your `application.conf`. Please note that this option is only available if you are not using delegated event processors. For more details, see the HTM event processor docs
- Added HTM Task Purging feature to allow purging items from `task` and `task-history` collections. TTL Indexes are created by default for both MongoDB and Azure CosmosDB for MongoDB, and they are needed for purging. Please look at HTM Task Purging documentation for more details on how to configure HTM for supported DBs.

Changed

- Global States (ACCEPTED, CANCELLED) removed from the Task Manager flow. This has been done in order to prevent ODS (Inquiry) marking a UnitOfWork as complete when actually it's not.

Bank Filtering

New

- Added filtering by ncc for check payment risk endpoint
- Added filtering by currency for check payment risk endpoint

Working Day Service

Fix Spotlight

- Fixes to the calculation of settlement and execution date (*PAY-11424*)

Changed

- Updated antora documentation for Working Days Service

Optional Modules - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-3-0/release-IPF-2024-3-0-aom.html

Optional Modules - Changes & Fixes

This page covers the optional module changes and fixes provided in release IPF-2024.3.0.

Scheme Packs

New

- Added documentation on how to configure SWIFT AGI Gateway
- SEPA Direct Debit Creditor MVP - support for IDF production and processing of DVF and RSF files
- New Collect and Settle API published

Changed

- The IP SIC Scheme Pack Inbound processing has been updated to no longer expect a pacs.002 (CNC002) confirmation message in response to a pacs.002 (NEG002) rejection of a pacs.008 message, and no longer sends pacs.028 chasers following the rejection.
- TIPS scheme message in validation requests now correctly returned in XML format instead of JSON
- Aligned messages produced by TIPS scheme pack to follow namespace prefixes convention as provided in the samples provided by the scheme (validation messages and payment messages)

SEPA CT CSM

New

- VFG simulator supports reading/writing to S3
- VFG simulator supports generating DVF and RSF files (RSF files for SEPA DD only)

Changed

- Scheme message in validation request messages are now correctly returning the modified message with transactions removed that are not valid
- Fixed ICF and IQF namespaces on produced files
- Sepa CT simulator, including docker images, have been renamed from sepact-vfg to sepa-vfg
- Extension points renamed from inbound/outbound - from scheme/to scheme

Bulker & Debulker - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-3-0/release-IPF-2024-3-0-bulker.html

Bulker & Debulker - Changes & Fixes

This page covers the Bulker & Debulker module changes and fixes provided in release IPF-2024.3.0.

Bulker

New

- Added the ability for child bulks to have the parent namespace prefix added when the bulk is being produced.

Debulker

New

- Added housekeeping settings to configure how to clean up the component store and source bulk file dependent on debulking outcome, i.e. Successful Processing, Failed Processing or Rejected. This is not a breaking change as the default settings are the same as previous versions.
- Added support for deletion of bulks from S3

CSM Reachability - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-3-0/release-IPF-2024-3-0-csm-reach.html

CSM Reachability - Changes & Fixes

This page covers the CSM Reachability V2 improvements, changes and fixes provided in release IPF-2024.3.0.

CSM Reachability and Industry Data Ingestion

Changed

- During file ingestion,file names and extensions are not case-sensitive anymore.

Data Ingestion

Changed

- Due to bug fixes in the connector-file library, all `ipf.csm-reachability.*.file-ingestion.interval` configuration properties should now be set in seconds, rather than minutes or hours. The bug fixes ensure that ingesters will no longer mistakenly pick up the same file during each polling cycle while it is being processed.
- CSM Reachability now includes default directory structures for the ingestion of supported industry data files. Each industry file now has a default, dedicated path, removing the requirement of each deployment to provide their own overrides. Default ingestion paths can still be customized if needed, existing overrides will still apply. For more details, refer to the **CSM Reachability Data Ingestion Configuration Reference** page.
- Default `ipf.csm-reachability.*.file-ingestion.interval` is reduced, from `1h` to `30s`. All existing `file-ingestion.interval` overrides can now be safely removed.
- Directory mapping from MongoDB `directory-mapping` collection is deprecated. It requires migration to `ipf.file-ingestion.directory-mapping` HOCON configuration that should be used instead. Please ensure you follow the appropriate instructions here.

Dynamic Processing Settings v2 - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-3-0/release-IPF-2024-3-0-dps-v2.html

Dynamic Processing Settings v2 - Changes & Fixes

This page covers the DPS (Dynamic Processing Settings) v2 module changes and fixes provided in release IPF-2024.3.0.

Dynamic Processing Settings v2

Dynamic Processing Settings API

New

- Implemented client library for DPS. DPS has generic API and with this client library DPS provides all applications to use it. It is implemented for CRUD and Search APIs. There are two ways how they can be implemented: via connectors or directly. When we use it via connectors we need to set client-type to be connector (that is also default value for connectors). If you want to use it directly, you should set client-type to direct (which is also default value when we are using direct approach).
- Added /setting-objects/schemas endpoint for getting payload and searchable fields schemas for all available setting objects. Also implemented client connectors for it.
- SettingsTypeIdResolver used for custom JSON (de)serialization. Response contains *@type* field which contains information about generic type instead of having *className* field with fully qualified class name. All settings have to set *@SettingTypeId* annotation with unique value. This value is set as *@type* field in JSON response.
- `dynamic-processing-settings-client-notification-port` module that represents an interface for receiving dps notifications
- `dynamic-processing-settings-client-notification-kafka` module that represents implementation for receiving dps notifications via kafka
- `DpsCrudNotification` type in `dynamic-proessing-settings-model`
- Error handling is compatible with DPS V1
- Approval API response contains new fields (approvalType, recordType, createdDate)
- deletedBy query parameter to the DELETE Setting API and the corresponding field on SettingDTO
- processingEntity as an optional query parameter for GET Approvals API

Changed

- Get approvals endpoint response extended to include the differences between persisted settings and approved settings.
- Approval Document (DB) structure has changed, old fields (approvedBy, rejectedBy, approvedReason, rejectedReason) are replaced by new fields (createdDate, resolvedBy, resolvedReason, resolvedDate, resolvedStatus)

Configuration

ipf.dps-api.client-type=connector	Client library implementation is done via connectors
ipf.dps-api.client-type=direct	Client library implementation is done via direct implementation
ipf.dps-api.default-connector	Default values for connectors
ipf.dps-api.default-connector.resiliency-settings	Default values for resiliency settings
ipf.dps-api.default-connector.resiliency-settings.max-attempts	Determines the maximum number of retries to be made. Note that this includes the first failed attempt. Default is set to 2.
ipf.dps-api.default-connector.resiliency-settings.retryable-status-codes	Retry if HTPP error code is in the list. Default values are [500, 503]
ipf.dps-api.default-connector.resiliency-settings.minimum-number-of-calls	Determines the minimum number of calls (within a sliding window period) that need to be made before the circuit breaker can calculate the error rate to determine the transport health
ipf.dps-api.connector.create-setting	Custom settings for create-setting connector. If not specified, default values are set.
ipf.dps-api.connector.update-setting	Custom settings for update-setting connector. If not specified, default values are set.
ipf.dps-api.connector.delete-setting	Custom settings for delete-setting connector. If not specified, default values are set.
ipf.dps-api.connector.get-setting	Custom settings for get-setting connector. If not specified, default values are set.
ipf.dps-api.connector.search-setting	Custom settings for search-setting connector. If not specified, default values are set.
ipf.dps-api.connector.search-ids-setting	Custom settings for search-ids-setting connector. If not specified, default values are set.
ipf.dps-api.connector.search-request-setting	Custom settings for search-request-setting connector. If not specified, default values are set.
ipf.dps-api.connector.setting-schemas	Custom settings for setting-schemas connector. If not specified, default values are set.
ipf.dps-api.client.notification	Custom settings for kafka consumer for dps crud notifications

Dynamic Processing Settings

New

- CRUD service v1 backward compatibility.
- DPS v2 can work with settings created by DPS v1 without any DB migration. It is necessary to configure DPS v2 with information on how to convert the v1 data structure to v2 for each setting definition. Configuration is done via property ipf.dps.settings.<setting-definition>. More details can be found in Configuration
- Search implementation which can return currently active setting value or setting value which is scheduled to be active in the future. Which value will be returned depends on the query parameter *atTime*. If the *atTime* parameter points to a time when the current value is active, then the current value will be used. If the *atTime* parameter points to a time in the future when the scheduled value will be active, then the scheduled value will be used. This parameter is optional. If not set, the current time is used.
- dynamic-processing-settings-notification-service with api module `dynamic-processing-settings-notification-service-port` and kafka implementation `dynamic-processing-settings-notification-service-kafka` for crud operations notifications
- in the `dynamic-processing-settings-repository-mongo` there is DpsIndexCreator bean that creates indexes per setting type based on hocon config
- Added History service implementation
- deletedBy field to the Setting document

- processingEntity as an optional parameter for GET Approvals

Changed

- `activeFromDate` is the parameter available in both `create` and `update` setting requests. This value tells us when the setting is active from. In case when this value is not provided in a request, the service will set current date at the moment of either creating or updating the setting.

Configuration

ipf.dps.settings.<setting-type>.type-mapping.from-type	Defines which v1 setting-specific repository model should be mapped to v2 repository model e.g. com.iconsolutions.ipf.dynamicsettings.repository.DpsSampleSettings
ipf.dps.settings.<setting-type>.type-mapping.to-type	V2 repository model used for v1 backward compatibility (which is no longer setting-specific and is used for all v1 settings). For v1 settings, it should be set to this value com.iconsolutions.ipf.dynamicsettings.v2.repository.mongo.entity.SettingDocumentV1
ipf.dps.settings.<setting-type>.type-mapping.payload-type	Setting definition class (which can remain the same as it used to be) e.g. com.iconsolutions.ipf.dynamicsettings.domain.DpsSample
ipf.dps.notification-service	Settings for dynamic-processing-settings-notification-service-kafka

Backward Incompatibility

- *activeFrom* parameter is removed from search query. Its usage is replaced with *atTime* parameter.
- Get approvals endpoint response extended to include the differences between persisted settings and approved settings.
- SettingsTypeIdResolver set *@type* field which contains information about generic type instead of having *className* field with fully qualified class name.

ODS & GUI - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-3-0/release-IPF-2024-3-0-gui-ods.html

ODS & GUI - Changes & Fixes

ODS

Changed

ODS supports receiving batches of records from IPF Processing Data Kafka topics, resulting in fewer and larger database writes, improving overall processing throughput.

GUI

New

- **Bulk Execute Human Task:**
    - Added ability to execute tasks in bulk
    - Can see the status of any bulk executions
- **GUI Support for SCT Operator Journey:**
    - Create screen for viewing transactions in the context of a batch and see all transactions in a batch in a single view and view batches in the context of a bulk and see all batches in a bulk
    - Ability to navigate between the parent, children and grandchildren of the transactions
- **Participant details to be viewable by clients in the IPF GUI:**
    - Added a new page to search and view participant data from the GUI
    - Can search for participant data based on the appropriate CSM agent ID
    - Can view participant details on a tables based on the search form and see the details when clicked through
- **Bank Filtering - Service API improvements:**
    - Show API validation errors on the form
    - Include NCC and NCC country on the create and search forms
    - Update create form layout
- **Configuration changes:**
    - Added new permissions roles for bulk executing tasks and viewing bulk task details: HTM_B_C, HTM_B_R
    - Can set the number of max tasks to bulk execute at one time (1000 as default): ipf.business-operations.human-task-manager.htm-bulk-page-results
    - Configuration for bulk execute types which is an array of Bulk Task Types: ipf.business-operations.human-task-manager.htm-bulk-task-types
    - HTTP client host and port transport configuration, for call time-out and resiliency-settings added for participant dynamic processing setting:
        - *ipf.business-operations.processing-settings.participant.http.client.host*
        - *ipf.business-operations.processing-settings.participant.http.client.port*
        - *ipf.business-operations.processing-settings.participant.http.client.call-timeout*
        - *ipf.business-operations.processing-settings.participant.http.client.resiliency-settings*
    - Added new permissions role for viewing participant data: ROLE_DPS_P_R
    - Added a mds-response-fields.conf which determines how to render and layout the various MDS search page tables in MDS Search module on the GUI. This has to be done per MDS object that you want to be displayed on your MDS search page.
    - Allow for assertionConsumerServiceLocation within SAML2 authorization to be configurable: ipf.business-operations.auth.saml2.assertion-consumer-service-location

Changed

- **JWT token used for authorization needs to be reduced in size:**

    - Reduced the length of the role names
- **Improvements to GUI to Support Reachability:**
    - Can now see agent names on bank filtering pages
    - Fixed issue with 403 error when changing from one processing entity to another when on the agent settings page
- **Core Project Defects**
    - Fixed issue where Bulk Execute E2E tests were flaky
    - Intra entity modal no longer appears when switching between processing entities in generic processing settings
    - User now actually gets redirected to login page when session is expired
    - Domain event correctly displays the supporting data
    - HTM now filters by processing entity
    - Dates are sorted correctly on payment search results
    - PDS history no longer replaces any others that are open on summary details
    - Bulk to batch related summary now works correctly

Removed

- Removed redundant attribute 'allowedProcessingEntities' from the cookie: *ipf.business-operations.jwt.allowed-processing-entities-claim*

Migration Steps for IPF-2024.3.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-3-0/release-IPF-2024-3-0-migration.html

Migration Steps for IPF-2024.3.0

Migration Steps for Flow Generation

Response and Reason codes

Reason and Response code enums are now generated ONLY in the model that they are used within. This will lead to two potential changes:

- The existing core definitions of the 'AcceptOrReject' response codes and 'ISOReasonCodes' reason codes are now provided as standard implementations. This means that the packaging of these classes is now fixed and not model dependent. Hence any use of these classes will require the import declaration to change to:
    - com.iconsolutions.ipf.core.flow.domain.input.AcceptOrRejectCodes
    - com.iconsolutions.ipf.core.flow.domain.input.ISOReasonCodes
- If using multi-model solutions, ensure that only the copy generated in the original model is referenced within the code. Similar to the above this may require changing the import packaging.

Importing Other Models

Previously when importing other models into a DSL based solution this was achieved by adding a block into the 'mps' module such as:

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-dependency-plugin</artifactId>
  <version>3.1.2</version>
  <executions>
      <execution>
          <id>unpack-ipf-business-functions-plugin</id>
          <phase>initialize</phase>
          <goals>
              <goal>unpack</goal>
          </goals>
          <configuration>
              <artifactItems>
                  <artifactItem>
                      <groupId>__groupid of target mps model goes here__</groupId>
                      <artifactId>__solution name of mps model goes here__</artifactId>
                      <version>${icon-business-functions-aggregator.version}</version>
                      <type>zip</type>
                      <overWrite>true</overWrite>
                      <outputDirectory>${plugin_home}</outputDirectory>
                  </artifactItem>
              </artifactItems>
          </configuration>
      </execution>
```

Now the key change is that the artifactId fields is now populated with the constant 'mps' (the name of the module itself) rather than being the solution name of the project.

> Note that this change is ONLY applicable once the downstream solution being referenced has been upgraded to 2024.3.0 and is not dependent on the version of the consuming project.

Flow Migration

Once updated to 2024.3.0, please run any migrations that you are prompted to perform when opening the project.

Migration Steps for Connectors

Resiliency Settings

- `withResiliencySettings(ResiliencySettings resiliencySettings)` has been deprecated and has been replaced with `Function<ResiliencySettings, ResiliencySettings> resiliencySettingsCustomiser` the purpose of this is to make resiliency config available for connector operations api.
- Before

```
.withResiliencySettings(ResiliencySettings.builder()
                        .withMinimumNumberOfCalls(1)
                        .withMaxAttempts(3)
                        .withRetryOnSendResultWhen(s -> {
                        .withRetryOnSendResultWhen(outcome -> {
```

```
                            // will retry only in state 1
                            var response = ((DeliveryOutcome) s).getResponse();
                            var response = ((DeliveryOutcome) outcome).getResponse();
                            return FAILURE_REPLY_STRING.equals(response.getReceivedMessage().getMessage().getPayload());
                        })
                        .build())
```

- Now the resiliency config should be passed back to the customiser. For example:

```
.withResiliencySettingsCustomiser(settings -> settings.toBuilder()
                        .withMinimumNumberOfCalls(1)
                        .withMaxAttempts(3)
                        .withRetryOnSendResultWhen(s -> {
                        .withResiliencyConfig(settings.getResiliencyConfig())
                        .withRetryOnSendResultWhen(outcome -> {
                            // will retry only in state 1
                            var response = ((DeliveryOutcome) s).getResponse();
                            var response = ((DeliveryOutcome) outcome).getResponse();
                            return FAILURE_REPLY_STRING.equals(response.getReceivedMessage().getMessage().getPayload());
                        })
                        .build())
```

The resiliency config will automatically be created and passed as the settings argument for use elsewhere

Local Directory Connectors and transport

- `FileHealthCheckConfig` configuration can now be specified per individual file transport. This can be achieved by using `LocalDirectoryConnectorTransport.builder()` and either:
    - including `fileCheckConfig` configuration block in the main file transport config file, or
    - directly providing custom root path to `FileHealthCheckSettings create(ClassicActorSystemProvider actorSystem, String configRootPath)` and including it into the builder by calling `.withFileHealthCheckSettings(FileHealthCheckSettings settings)` method on the builder
- `LocalDirectoryConnectorTransport(ActorSystem actorSystem, String name, FileIngestionConfiguration fileIngestionConfiguration)` is deprecated, and it will be removed in the next release. Please use `LocalDirectoryConnectorTransport.builder()` instead
- `LocalDirectoryTransportConfiguration(String configRootPath, Config config)` is deprecated, and it will be removed in the next release. Please use `LocalDirectoryTransportConfiguration(ClassicActorSystemProvider actorSystem, String configRootPath)` instead
- `static FileHealthCheckSettings createDefault(Config config)` is deprecated, and it will be removed in the next release. Please use `static FileHealthCheckSettings create(ClassicActorSystemProvider actorSystem, String configRootPath)` instead
- `withTransportConfiguration` method on `LocalDirectoryConnectorTransport.Builder` is marked as deprecated and scheduled for removal
- `LocalDirectoryConnectorTransport` will now filter out files that are currently being processed from its polls, enabling `interval` to be safely set to durations shorter than expected processing times — seconds instead of hours.

Deprecating directory mapping from MongoDB `directory-mapping` collection

Directory mapping from MongoDB `directory-mapping` collection will be deprecated and moved to the `ipf.file-ingestion.directory-mapping` HOCON configuration that will be used for directory mappings. From now on, it's not allowed to have a file ingester without and appropriate `directoryId` in `directory-mappings`.

Migration steps

1. Backup all data from Mongo `directory-mapping` collection.
2. For each custom ingester ensure adding related Mongo document data from `directory-mappings` collection to ingesters' .conf file.
3. Restart application and check if there is no warnings in log with message `Missing required HOCON configuration: ipf.file-ingestion.directory-mappings.`
4. Make sure that log doesn't contain warnings like:
    a. `Mongo directory-mappings documents value doesn't exist in Hocon configuration.`
    b. `Mismatch found for Mongo directory-mappings documents value and Hocon configuration.`
5. Delete Mongo `directory-mapping` collection if previous steps are fullfiled.

Http Connectors and transports

- `HttpConnectorTransport<T>.Builder` should use only the name, actor system and config root path when building transports.
    - Use `<T> Builder<T> builder(String name, ClassicActorSystemProvider actorSystem, String configRootPath)` .
- `HttpReceiveConnectorTransportFactory` is deprecated and will be removed, so use `HttpReceiveConnectorTransport.Builder` instead.
- `withTransportConfiguration` method on `HttpConnectorTransport<T>.Builder` and `HttpReceiveConnectorTransport.Builder` is marked as deprecated and scheduled for removal
- Use `status-codes-treated-as-errors` to define status codes that are errors and can't be ignored. These status codes will be use while building treatErrorResponseAsFailureWhen predicates.
- Use `<REQ_D, REQ_T, REP_D, REP_T> Builder<REQ_D, REQ_T, REP_D, REP_T> builder(String name, String configRootPath, ClassicActorSystemProvider actorSystem)` when building Request-Reply Send connectors.

JMS Connectors and transports

- JMS Connector Transport builder should use only name, actor system, config root path and connection factory.
- `JmsConnectorTransportFactory` is deprecated and will be removed, so use `JmsConnectorTransport.Builder` instead.
- `JmsReceiveConnectorTransportFactory` is deprecated and will be removed, so use `JmsReceiveConnectorTransport.Builder` instead.
- `withTransportConfiguration` method on `JmsAckReceiveConnectorTransport.Builder` , `JmsConnectorTransport.Builder` and `JmsReceiveConnectorTransport.Builder` is marked as deprecated and scheduled for removal

Kafka Connectors and transports

- When building String-String Kafka transports, `KafkaConnectorTransport` , `KafkaReceiveConnectorTransport` and `KafkaAckReceiveConnectorTransport` , use stringBuilder and provide only name, actor system and config root path.

Migration Steps for Icon Akka Plugins

Akka Discovery MongoDB

`akka.discovery.akka-mongodb.uri` , `akka.discovery.akka-mongodb.set-ssl-context` and `akka.discovery.akka-mongodb.ssl-context` will now default to their `ipf.mongodb` counterparts (`ipf.mongodb.url` , `ipf.mongodb.set-ssl-context` and `ipf.mongodb.ssl-context` , respectively) and no longer have to be manually set if the counterparts are provided.

Akka Lease MongoDB

`akka.coordination.lease.mongodb.url` , `akka.coordination.lease.mongodb.set-ssl-context` and `akka.coordination.lease.mongodb.ssl-context` will now default to their `ipf.mongodb` counterparts (`ipf.mongodb.url` , `ipf.mongodb.set-ssl-context` and `ipf.mongodb.ssl-context` , respectively) and no longer have to be manually set if the counterparts are provided.

Akka Persistence MongoDB

`iconsolutions.akka.persistence.mongodb.read-concern` has been removed, use `readConcernLevel` option in the connection string to set the read concern.

`iconsolutions.akka.persistence.mongodb.url` , `iconsolutions.akka.persistence.mongodb.set-ssl-context` and `iconsolutions.akka.persistence.mongodb.ssl-context` will now default to their `ipf.mongodb` counterparts (`ipf.mongodb.url` , `ipf.mongodb.set-ssl-context` and `ipf.mongodb.ssl-context` , respectively) and no longer have to be manually set if the counterparts are provided.

Migration Steps for IPF Processing Data Version 2

All core IPF applications are able to consume data from both the V2 and V1 IPF Processing Data model. By default, all IPF Processing Data Egress plugins will export data using the V2 data model. If you have any custom applications that consume from IPF Processing Data, the following steps should be taken.

Set Egress Applications to use V1

Your consuming applications cannot handle the V2 data model, therefore for now you should continue to export using the V1 data model. For all applications that utilise the IPF Processing Data Egress plugins, configure `ipf.processing-data.egress.schema-version = 1` to continue to produce data using the V1 data model.

Update consuming applications

Update each application that consumes from IPF Processing Data so that they can handle both the V2 and V1 data model. For more information, see the consume IPF Processing Data guide.

Set Egress Applications to use V2

Once all your consuming applications are able to handle both the V2 and V1 data model, you can safely update your producers to export messages using the V2 data model. This can be done by configuring `ipf.processing-data.egress.schema-version = 2`

Event Processor ID Resolution Fix in Egress Journal Processor

Issue Overview

In versions prior to 2024.3, the `ipf-processing-data-egress-journal-processor` module had an inconsistent resolution of the `event-processor.id` configuration property. Depending on Java classpath resolution at runtime, the event processor ID would resolve to either `EventProcessor` (incorrect value) or `IpfProcessingDataEventProcessor` (intended value).

Resolution

Version 2024.3 fixes this inconsistency. The event processor ID will now correctly resolve to IpfProcessingDataEventProcessor in all cases.

Migration Impact

Services that previously resolved to `EventProcessor` require configuration changes during the 2024.3 upgrade to prevent their Egress Journal Processors from reprocessing the entire event journal.

Determining If Action Is Required

> You need to perform this check for each orchestration application, otherwise you risk issues in production.

You can verify if your application needs configuration changes using either of these methods.

For users with network access to service instances, run:

```
curl -s localhost:8080/actuator/info \
  | grep -o -P '"event.processor.id":"EventProcessor"'
```

For users with MongoDB access:

```
mongo <connection_params_omitted> --eval\
 'db.mongoOffset.find({"_id.eventProcessorId":"EventProcessor"})'
```

Required Configuration Change

If either verification method returns results, add the following line to your service's `application.conf` file:

```
event-processor.id = EventProcessor
```

Release Notes for IPF-2024.2.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-2-0/release-IPF-2024-2-0.html

Release Notes for IPF-2024.2.0

This page details everything required to get going on IPF Release 2024.2.0 made available on **21st August 2024**.

Change Spotlight

> - Changes to MPS Build scripts are required. If you are using buildscripts in your project please ensure you follow the appropriate instructions here

> - Archetype has now been replaced by **ipf-project-scaffolder**
>
> The latest version of the scoffolder is **1.0.3.1**
>
> Instructions on how to use the scaffolder to create new projects are here

- All external MPS language dependencies are now captured in a single project, providing less future change to client code.
- IPF Usage Report Endpoint added `actuator/ipflicense`, this report may periodically be requested by IPF Support

Fix Spotlight

- Fixed issue where receive connector transport message ack not sent if an unexpected exception thrown during handling of a received message that cannot be parsed - (*PAY-10957*).

Release Details

Binaries

The core binaries are available from IPF-Releases.

Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Akka/Alpakka binaries from 2.9.x onwards are no longer on Maven Central and are available from the Akka Library Repository which Icon mirror.

To enable the new mirror, add the following to your settings.xml in the `<repositories>` section:

```
<repository>
  <id>akka-repository</id>
  <name>Akka library repository</name>
  <url>https://nexus.ipf.iconsolutions.com/repository/akka-repository</url>
</repository>
```

Some MPS dependencies are also mirrored by Icon and can be found by adding the following repository to the same section:

```
<repository>
  <id>icon-thirdparty</id>
  <name>IPF icon-thirdparty repository</name>
  <url>https://nexus.ipf.iconsolutions.com/repository/icon-thirdparty</url>
</repository>
```

Add the following to your `<servers>` section for each new repository:

```
<server>
  <id>Reposity name here</id>
  <username>xxxx</username>
  <password>xxxx</password>
</server>
```

Where `xxxx` is the username and password repectively provided to you by Icon.

Developer App

The latest version of the Developer App is **2.3.8**

2024.2.0 Jar/Pom list

The Jars and Poms for 2024.2 and the associated versions are listed here: 2024-2-artifacts.xlsx

Core - Improvements, Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-2-0/release-IPF-2024-2-0-core.html

Core - Improvements, Changes & Fixes

This page covers core improvements, bug fixes, changes and fixes provided in release IPF-2024.2.0.

Core improvements, bug fixes & changes

New

- **IPF Platform**: Journal processor documentation added to Flo Starter Projects
- **Connector**: Configuration property added for timing out call to the `CorrelationService` in `SendConnector`. Default value provided at `ipf.connector.default-send-connector.correlation-stage-timeout` of 5s.

  > If the value provided for `correlation-stage-timeout` is not less than the `call-timeout` specified for a particular connector the correlation stage timeout will be reduced to be less than the call timeout (200ms less) and the updated value will be logged alongside a warning message.

- **MongoDB**: Added capability to set the commit quorum. Can be globally set for all index creation with `ipf.mongodb.commit-quorum`. See individual component documentation for how to override per-component.
- **IPF Processing Data**: Exporters can produce data to different Kafka topics, configurable per data type.
- **Flo-lang and Akka-persistence-mongo-db**: Added configurable purging functionality for the journal and snapshot collections. Default functionality it to not purge documents from either collection. Implementation utilises Mongo and Cosmos ttl indexes which will need to be created manually. Configuration guides found in docs:
  - Flo-lang Docs
  - Persistence Plugin Docs
- **Persistent Scheduler** added timezone support to persistent scheduler

Changed

- **IPF File Poller** - Breaking change - To support multiple processing entities the IPF File Poller can now poll from multiple locations. This means the following config has now changed from a single item to a list of items: `ipf.file-poller` → `ipf.file-poller.pollers`
- **Dynamic Settings Workflow** - Redundant call to file converter during file ingestion was removed from file processor. This issue was affecting CSM Reachability Data Ingestion: FileEntrySkippedEvent and PartyEntityDirectorySubTypeMappingSkippedEvent file processing events were raised twice and errors were logged twice.
- **Dynamic Settings Workflow** - Added new event and enriched existing events. It will improve monitoring, for already existing `FileEntrySkippedEvent` with type and fileName, `ProcessingCompleteEvent` with type, outcome, file_name and process_name and `ProcessingFailedEvent` with processName and fileName. Also added new `FileEntryProcessedEvent` with metrics type and fileName(more about it in Csm Reachability and Industry Data ingestion)
- Replacing Caffeine sync cache implementation with async cache implementation to fix multiple calls to callback in `getOrDefault` method in `ipf-cache-caffeine` module.
- Updated `EventProcessorStream` to use `mapAsyncPartitioned` instead of the previous `mapAsync` operator. Now, even when processing parallelism is enabled we won't be processing related events in parallel within a single stream, thus journal processors to be safely parallelised.
- Ipf-file-manager - S3FileReader fixed to be able to download bigger files.
- **Connector**: `IngestedFile` as `ReceivedMessage's receive context has been replaced with `IngestedFileContext`
- **IPF Processing Data**: Updated `MdsWrapper` class's generic type constraint. The generic parameter `T` must now implement `java.io.Serializable`
  - Changed from `MdsWrapper<T>` to `MdsWrapper<T extends Serializable>`
- **IPF Transaction Cache** - changed names of indexes which are being created on transactionCacheEntry mongodb collection:
  - `findByTypeAndHashIndex` renamed to `hash_1_type_1`
  - `findByTypeAndHashAndMessageIdIndex` renamed to `hash_1_type_1_messageId_1`

Fixed

- **Connector** - Memory consumption in `LocalDirectoryConnectorTransport` component
- **IPF Archiver** - Fixed an issue where errors delivering archive bundles to Kafka were not correctly propagated, resulting in missing archive bundles.

Configuration

Deprecated `event-processor.upstream-event-demand`	Backward compatibility is maintained for this release but `event-processor.processing-parallelism` should be used instead
Deprecated `stream.processor.cluster-role`	Backward compatibility is maintained for this release but `event-processor.cluster-role` should be used instead
Introduced `event-processor.event-streaming-type` configuration property	Defaults to `EVENT_STREAM_PER_TAG`, which represents the previous behaviour that relies on user-provided list of global tags. Before switching an existing system to use `EVENT_STREAM_PER_FLOW` — required for rolling upgrades of your orchestration services — please ensure you have performed the necessary database migrations (please refer to the migrations). Failing to do so will effectively reset the offsets in your journal processors, causing them to reprocess every event in your journal.
Introduced `ipf.processing-data.egress.transports.(events|data-structures|message-logs|system-events|models|custom-objects)`	These transports default to existing Kafka transport configuration, with the default `IPF_PROCESSING_DATA` topic. Kafka clients, and by extension the topics, can be configured per data type, e.g. all message logs can go to a different topic, e.g. `ipf.processing-data.egress.transports.message-logs.kafka.producer.topic = MESSAGE_LOG_TOPIC`. This change is non-breaking and behaves as before unless explicitly configured to use different topics.

Optional Modules - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-2-0/release-IPF-2024-2-0-aom.html

Optional Modules - Changes & Fixes

This page covers the optional module changes and fixes provided in release IPF-2024.2.0.

Scheme Packs

New

- Added IPF gateway headers to TIPS message sent to scheme (configurable)
- Character replacer can replace parts of the message

Changed

- The TIPS pacs.008 duplicate check now allows setting the inbound and outbound message duplicate check expire period and the cache entries purge.
- TIPS scheme pack to use the revised November 2024 schemas
- SIC5 scheme pack updated to use new IBAN Deconstruct response from CSM Reachability.
- ID generation in scheme packs to be as follows:

```
UnitOfWorkId = UUID
AssociationId = UUID (different to above UUID)
ClientRequestID :
    pacs.008 = TxID
    camt.056 = Undrlyg.TxInf.CxlId
    pacs.004 = TxInf.RtrId
    camt.029 = CxlDtls.TxInfAndSts.CxlStsId
```

- Additional validation added to TIPS inbound/outbound pacs008 processing in scheme pack
    - Total Interbank Settlement Amount
    - Organisation Identification Party Subfields
    - Private Identification Party Subfields
    - Postal Address subfields
    - Remittance Information Subfields
- Additional validation added to TIPS outbound pacs.004
    - Only single transaction present
    - OrgnlGrpInf occurs only once
    - Total Returned Interbank Settlement
    - Technical Duplicate Check
- Additional validation to outbound pacs.002
    - OrgnlMsgNmId validation
    - GrpSts/TxSts mutually exclusive
    - StsRsnInf populated for negative pacs.002
- Additional validation to outbound pacs.028
    - OrgnlMsgNmId is valid
    - AccptDtTm mandatory
    - Amount field mandatory

Fixed

- TIPS Simulator not setting both group header amount and transaction amount when overridden
- SIC does not cancel pacs.028 when TM01 received

Configuration changes:

- Added:
  - `tips.processing-entity.pacs008-duplicate-message-check-period` to set pacs.008 message expire period for the inbound and outbound message duplicate check
  - `tips.processing-entity.pacs008-purging-cache` to enable and set purging policy for duplicate checks cache entries
- Removed:
  - `tips.rule.outbound-bicfi` - Clients who used this configuration in the TIPS service must assure that the BIC set for the removed configuration is in the list of approved debtor agent BIC specified in the `tips.rule.allowed-bics.dbtragt` configuration. This change align TIPS with other CSM packs.
- Changed:
  - Default configuration has been added to fetch configuration values for `ToScheme` `ClearAndSettleRequest` and `SendPositiveAnswerToCSMRequest`, with the default value set to `false`. The extending schemas have been updated to utilize these configuration values while ensuring backward compatibility.

CSM	Config Value Supported	Config Value Deprecated
Tips	`ipf.csm.tips.pacs008.instg-agt-id`, `ipf.csm.tips.pacs008.instd-agt-id`, `ipf.csm.tips.pacs008.map-instd-agt-from-config-enabled`	`tips.instg-agt-bic`, `tips.instd-agt-bic`, `tips.bics-from-config-enabled`
RT1	`ipf.csm.rt1.pacs008.instg-agt-id`, `ipf.csm.rt1.pacs008.instd-agt-id`, `ipf.csm.rt1.pacs008.map-instd-agt-from-config-enabled`	`rt1.instg-agt-bic`, `rt1.instd-agt-bic`, `rt1.bics-from-config-enabled`
SIC	`ipf.csm.sic.pacs008.instg-agt-id`, `ipf.csm.sic.pacs008.instd-agt-id`, `ipf.csm.sic.pacs008.map-instd-agt-from-config-enabled`	
T2	`ipf.csm.t2.pacs008.instg-agt-id`, `ipf.csm.t2.pacs008.instd-agt-id`, `ipf.csm.t2.pacs008.map-instd-agt-from-config-enabled`	`t2.instg-agt-bic`, `t2.instd-agt-bic`
FedNow	`ipf.csm.fednow.pacs008.instg-agt-id`, `ipf.csm.fednow.pacs008.instd-agt-id`, `ipf.csm.fednow.pacs008.map-instd-agt-from-config-enabled`	

IPF Metrics Processor

Changed

- Metrics produced by the IPF Metrics Processor have been renamed to be prefixed with `_ipf`, but this is initially **opt-in** with the configuration property `ipf.business-metrics-processor.opt-in-to-ipf-prefixed-metric-names = true`.
- Configuration of metrics and labels now supports dot-notation. The configuration structure has changed slightly to accommodate this.
- Payment metrics are now labelled with the processing entity when it is known.
- Payment metrics are now labelled with the result of a creditor comparison, if one was performed during the payments lifecycle.
- Payment metrics are now labelled with the most recent error code.
- Introduced a payment waiting duration metric representing the time payments spend in waiting states.

Bulker & Debulker - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-2-0/release-IPF-2024-2-0-bulker.html

Bulker & Debulker - Changes & Fixes

This page covers the Bulker & Debulker module changes and fixes provided in release IPF-2024.2.0.

Bulker

Changed

- Breaking change - File path and archiving path have now moved to specific configurations to support multiple processing entities and output directories
    - The following configuration items have been moved from → to (Where [0] represents a location in the array of config items):
        - ipf.bulker.producer.file-path → ipf.bulker.configurations[0].file-path
        - ipf.bulker.archiver.file-path → ipf.bulker.configurations[0].archive-path
- Breaking change - If using the S3 BulkOutputStreamProvider the following configuration has moved from → to (Where [0] represents a location in the array of config items):
    - s3.config.bucket → ipf.bulker.configurations[0].file-path
- The BulkOutputStreamProvider bean is no longer required to be created and will be auto configured by bulker

Fixed

- NullPointerException on rehydration fixed
- File production error can cause loop on bulk close and finalising
- Auto close scheduler fails to schedule jobs if deployed in timezone ahead of UTC

Debulker

Changed

Processing entity and archiving path have now moved to specific configurations to support multiple processing entities and input directories

- The following configuration items have been moved from → to (Where [0] represents a location in the array of config items):
    - ipf.debulker.processing-entity → ipf.debulker.configurations[0].processing-entity
    - ipf.debulker.archiver.file-path → ipf.debulker.configurations[0].archive-path

---

CSM Reachability - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-2-0/release-IPF-2024-2-0-csm-reach.html

CSM Reachability - Changes & Fixes

This page covers the CSM Reachability V2 improvements, changes and fixes provided in release IPF-2024.2.0.

CSM Reachability and Industry Data Ingestion

New

- `csm-reachability-file-ingestion` with the `csm-reachability-file-ingestion-api` module and the support for ingesting files from aws s3 bucket (`csm-reachability-file-ingestion-s3`).
    - Topic name to send file available notification is `FILE_AVAILABLE_NOTIFICATION`
    - Topic name to receive file processed notification is `FILE_PROCESSED_NOTIFICATION` The purpose of this is that we are now able to read and process files located on an aws s3 bucket instance.
- `csm-reachability-file-ingestion-notification-service` module with `csm-reachability-file-ingestion-notification-service-core` containing message types and kafka support (`csm-reachability-file-ingestion-notification-service-kafka`). It is possible now to receive a kafka message saying the file is available for processing on the s3 bucket, trigger the file processing mechanism, and send the message to kafka if needed, saying the file has been processed (successfully or not).
- Send Connector for sending `FileProcessedNotification` message to a specified Kafka topic
- Receive Connector for receiving `FileAvailableNotification` message
- `FileAvailableNotificationHandler` for handling FileAvailableNotification, triggering file processing and optionally acknowledging whether processing is successful or failed.
- `FileEntryProcessedEvent` is raised for each file entry processed.

Changed

- In `ProcessingFailedEvent` `name` field is renamed to `processName` , because now we have field called `fileName` , and this `processName` is more descriptive than just `name` .
- Bank Directory Plus load to Party Entity correctly maps `identifierSubType` when proper correlation between `iso_country_code` and `field_b` exists in input file. Mapping is configurable with property `ipf.csm-reachability.data-ingestion.bank-directory.mappings`
- Absorbed `connector` and `flo-lang` changes which will together with configuration changes improve memory and time consumption during file ingestion.
- When csm-reachability data ingestion fails, the message level is now error (was previously info) and the message includes the filename.

Configuration

`ipf.csm-reachability.file-ingestion-notification-service.directory-ids`	directory id config properties for ingested files from s3 bucket on path
`ipf.csm-reachability.data-ingestion.bank-directory.mappings`	mapping from field_b to identifierSubType for Bank Directory Plus load to Party Entity
`ipf.csm-reachability.file-ingestion.s3.enabled`	property to enable S3 file ingestion. Default value is false.
`ipf.file-manager.s3.resiliency-settings.max-attempts`	Determines the maximum number of retries to be made. Note that this includes the first failed attempt. Default value is 2.
`ipf.file-manager.s3.resiliency-settings.retryable-status-codes`	Retry if HTTP error code is in the list. Default value is [500, 503]
`ipf.file-manager.s3.resiliency-settings.attempt-timeout`	Default value is 2s.
`ipf.file-manager.s3.resiliency-settings.call-timeout`	Default value is 3s.
`ipf.file-manager.s3.endpoint-url`	S3 endpoint url
`ipf.file-manager.s3.region`	S3 region
`ipf.file-manager.s3.credentials.access-key-id`	S3 Credentials
`ipf.file-manager.s3.credentials.secret-access-key`	S3 Credentials
`ipf.file-manager.s3.path-style-requests`	Path style request

CSM Reachability API

New

- V2 API introduced with prefix `/api/v2`
- Previous version of API is no longer supported
- Swagger UI is available at url host:port/swagger-ui/index.html
- Swagger UI updated with valid descriptions and examples
- New endpoint `/v2/validate-intra-entity-reachability` added which is used to verify if counter-party is reachable via processing entity (OnUs) so there is no need to send transfer via external agent or CSM.
- Added new attribute entityCountry in IbanServiceResponseDto for /V2/iban-deconstruct response. It is mapped from the attribute isoCountryCode on IBAN Plus Dynamic configuration.
- Api connector configuration is added, and it follows the ipf naming standard. Each connector inherits from default connector configuration and can override it for its own purpose. Default configuration is under `ipf.csm-reachability-api.default-connector`, and for each existing connector, configuration is under `ipf.csm-reachability-api.connector.<connector-name>`
- New enum value PARTICIPANT_COUNTRY for appliedLimitCategory in ValidateScmReachability and SelectCsmAgent responses.
- participantCountryLimits in AgentSettlementSettings response
- new field named `agentCountry` as a part of AccountServicingAgent in Agent Settlement Settings

Changed

- Api connector transport configuration is now changed to match the ipf naming standard, but is still backward compatible with the old naming.
- Configuration property prefix `csm-reachability-api` is changed to `ipf.csm-reachability-api`
- For `/party-entities` GET endpoint `entityDataSource` parameter is now mandatory, required on GetPartyEntitiesCriteria.
- When identifier is BIC, identifierSubType will be ignored during the Party Entity LookUp

- IbanServiceResponseDto is changed to support multiple entityIdentifiers. Removed ibanBic, sortCode and headOfficeIid from the response, and added ibanCountry and list of entityIdentifiers (BIC or NCC).
- Iban deconstruct can handle countries that don't issue national ids

Configuration

`ipf.csm-reachability-api.default-connector`	default connector configuration
`ipf.csm-reachability-api.connector.<connector-name>`	configuration of a specific connector

CSM Reachability

New

- CsAgentSelectionSettings - added selectionId as a searchable field
- IbanPlus - added new attribute isoCountryCode. Logical unique key now is updated and it's consisted of isoCountryCode + ibanIsoCountryCode + nationalId fields.
- Dps connector configuration is added, and it follows the ipf naming standard. Each connector inherits from default connector configuration and can override it for its own purpose. Default configuration is under `ipf.csm-reachability.settings-api.default-connector`, and for each existing connector, configuration is under `ipf.csm-reachability.settings-api.connector.<connector-name>`.
- When the Party Entities API is called with a specific data source, multiple records can now be retrieved. In this scenario, we will return a new reason code with the description: `Multiple party entity records match on Party Entity Directory for specified identifier value. Reachability request can be re-run including the identifier type used for membership by the Agent for accurate results`.
- participantCountryLimits - new type of limit se on AgentSettlementSettings related to participant country
- DPS - new field named `agentCountry` as a part of AccountServicingAgent in Agent Settlement Settings. New field is mandatory when agent identifier type is NCC or LEI
- when /select-csm-agent endpoint is called and in the request we have returnEnhancedCSMData as true, agentCountry is mapped into the response. Also, when /settlement-agent endpoint is called and isIncludeSettlementSettings as a true in the request, agentCountry is mapped into the response.
- New `AgentCountryPattern` which doesn't allow that new field `agentCountry` is not set when we have at least 1 agent identifier type which is NCC or LEI

Changed

- All properties, related to setting save history, are changed to match IPF standard, but are still backward compatible with the old naming. Old naming is deprecated and scheduled for removal in future releases.
- Dps connector transport configuration is now changed to match the ipf naming standard, but is still backward compatible with the old naming. Deprecated configuration will be scheduled for removal in future releases.
- When matching entity identifiers on party entity lookup, `identifierSubType` is used together with `identifierType` if supplied
- IbanServiceResponseDto is changed to support multiple entityIdentifiers. Removed ibanBic, sortCode and headOfficeIid from the response, and added ibanCountry and list of entityIdentifiers (BIC or NCC).
- EntityCountry is mapped as ISO Country code from Iban Plus record.
- IbanCountry is mapped as IBAN ISO Country code from Iban Plus record.
- BIC value is now mapped as one of the identifiers if we have ibanBic on the ibanPlus record, and the subtype is mapped as SCHEME_MEMBERSHIP_BIC.
- NCC value can be mapped as IBAN National ID or as IBAN National ID (without the 4 character SWIFT Bank Code), and the subtype is mapped if the ibanCountry has corresponding ISO Country Code mapping.
- Party entity service is now using an IBAN deconstructed identifier with BIC type if the entityDataSource on the criteria is Bank_Directory_Plus, otherwise it is using NCC identifier for look-up.
- Validate intra entity reachability service is using an IBAN deconstructed identifier with BIC type if it is specified on IntraEntityParties record, otherwise it is using NCC identifier as CounterPartyIdentifier.
- Validate csm entity reachability service is using an IBAN deconstructed identifier with BIC type if it is specified on ParticipationDetails on AgentSettings record, otherwise it is using NCC identifier as CounterPartyIdentifier.

- Validate CSM Reachability and Select CSM Agent - participant country match removed from participant lookup

Configuration

`ipf.csm-reachability.settings-api.default-connector`	default connector configuration
`ipf.csm-reachability.settings-api.connector.<connector-name>`	configuration of a specific connector
`ipf.csm-reachability.settings-api`	changed from `settings-api`
`ipf.csm-reachability.settings-api.connection`	changed from `settings-api.connection`
`ipf.csm-reachability.should-save-history.agent-clearing-settings-settings`	changed from `should-save-history.agent-clearing-settings-settings`
`ipf.csm-reachability.should-save-history.agent-settings-settings`	changed from `should-save-history.agent-settings-settings`
`ipf.csm-reachability.should-save-history.agent-settlement-settings-settings`	changed from `should-save-history.agent-settlement-settings-settings`
`ipf.csm-reachability.should-save-history.bicdir2018-settings`	changed from `should-save-history.bicdir2018-settings`
`ipf.csm-reachability.should-save-history.cs-agent-selection-settings-settings`	changed from `should-save-history.cs-agent-selection-settings-settings`
`ipf.csm-reachability.should-save-history.exclusion-list-settings`	changed from `should-save-history.exclusion-list-settings`
`ipf.csm-reachability.should-save-history.generic-processing-settings-settings`	changed from `should-save-history.generic-processing-settings-settings`
`ipf.csm-reachability.should-save-history.iban-plus-settings`	changed from `should-save-history.iban-plus-settings`
`ipf.csm-reachability.should-save-history.iban-structure-settings`	changed from `should-save-history.iban-structure-settings`
`ipf.csm-reachability.should-save-history.participant-settings`	changed from `should-save-history.participant-settings`
`ipf.csm-reachability.should-save-history.party-entity-settings`	changed from `should-save-history.party-entity-settings`
`ipf.csm-reachability.should-save-history.processing-entity-settings`	changed from `should-save-history.processing-entity-settings`

Data Ingestion

New

- IbanPlus file ingestion - isoCountryCode filed is loaded from IBANPlus file with the same name. Mapping is done for both txt and xml files, and also for full and delta files.

Changed

- Data ingestion configuration is now changed to match the ipf naming standard, but is still backward compatible with the old naming. Deprecated configuration will be scheduled for removal in future releases.

Configuration

`ipf.csm-reachability.connector.settings-api-rr`	inherits default `ipf.csm-reachability.default-connector` configuration
`ipf.csm-reachability.default-file-ingestion`	with default settings for file-ingestion which every file ingestion connector inherits
`ipf.csm-reachability.setting-api`	changed from `settings-api`
`ipf.csm-reachability.ingestion`	changed from `ingestion`
`ipf.csm-reachability.participant.tips`	changed from `tips` (file-ingestion-connector is removed from new path)
`ipf.csm-reachability.participant.rt1`	changed from `rt1` (file-ingestion-connector is removed from new path)
`ipf.csm-reachability.participant.step2`	changed from `step2` (file-ingestion-connector is removed from new path)
`ipf.csm-reachability.participant.sic`	changed from `sic`
`ipf.csm-reachability.party-entity.six-bankmaster`	changed from `party-entity.six.bankmaster` (file-ingestion-connector is removed from new path).
`ipf.csm-reachability.party-entity.swift-bankplus`	changed from `party-entity.swift.bankplus` (file-ingestion-connector is removed from new path)
`ipf.csm-reachability.iban-plus`	changed from `swift.ibanplus` (file-ingestion-connector is removed from new path)
`ipf.csm-reachability.exclusion-list`	changed from `exclusionlist` (file-ingestion-connector is removed from new path)
`ipf.csm-reachability.bic-dir-2018`	changed from `bic-dir-2018` (file-ingestion-connector is removed from new path)
`ipf.csm-reachability.iban-structure.enabled`	changed from `ibanstructure.process-ibanstructure.enabled`
`ipf.csm-reachability.iban-structure.file-ingestion`	changed from `iban-structure-file-connector`

ODS & GUI - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-2-0/release-IPF-2024-2-0-gui-ods.html

ODS & GUI - Changes & Fixes

This page covers the Operation Data Store (ODS) and GUI changes and fixes provided in release IPF-2024.2.0.

ODS

New

- V2 ODS Inquiry API specification.
    - New Inquiry API version introduced to remove all `/payment-objects` endpoints (replaced with `/mds-objects`) and to unify search parameters and response types of the `/views/summaries/` endpoints.
    - The ODS Inquiry application can handle calls to both the V2 and V1 Inquiry APIs
    - A new set of client connectors can be enabled for the V2 API.

> It is important to note that a config change **MUST** be applied if you are already using the V1 client connectors and do not want to immediately migrate to the V2 client connectors. Your configuration must be updated to override the client connector version configuration as follows: `ods.inquiry.client.version=1`
>
> This will ensure your current V1 client connectors remain enabled and a code change will not be required. See V1 Inquiry API client documentation for more information on the V1 client APIs that can be enabled.

- Versioning details and the full V2 migration guide can be found on the API versioning overview page

Changed

- Fixed an issue where ODS attempted to amend documents in the `unitOfWorks` collection concurrently, sometimes resulting as two inserts, and therefore a duplicate key exception. This impacted archiving which relied on attributes of the unit of work that had failed to be written.
- The following changes apply to ODS Inquiry API Version 2 only
    - Improved support for customised summary mappings in ODS where the customised fields are not typically used for the journey type being customised.
        - Simplified summary responses - all possible summary fields are returned for all journey types.
        - Simplified summary search parameters - all possible summary search fields are supported for all journey types.

GUI

New

- **GUI to reflect changes in Reachability data model:**
    - Updated ops-gui-service-ng with new reachability end points
    - Created new list card component
    - Implemented create page for Agent Settlement Setting
    - Implemented create page for Agent Clearing Setting
    - Added new multi select component to the dynamic form
    - Implemented edit pages for Agent Setting, Agent Settlement Setting and Agent Clearing Setting
    - Added new roles for the new Reachability modules
    - Added new time input to the dynamic form
    - Added Generic Processing Settings endpoint to dashboard
    - Linked Agent Setting to Bank Filtering
    - Implemented view page for Agent Settlement Setting
    - Shortened processing settings roles
    - Created a new loading spinner
    - Added new multiple text input to the dynamic form

- Used config rather than enums in Processing settings Module so the inputs are configurable by the user
- Implemented new approval/rejection modal
- Displaying validation errors from the API in the various forms across Agent Settings and Generic Processing Settings
- New HTM screen for outbound Return awaiting scheme response
- GUI to support Creditor Reference as a conditional field
- Enable Audit for ODS Export

- **Configuration changes:**
  - HTTP client host and port transport configuration, requiring of approval for deletion, creation and updating, call time-out and resiliency-settings added for agent settings, agent clearing settings, agent settlement settings and generic processing settings:
    - *ipf.business-operations.processing-settings.agent-settings.http.client.host*
    - *ipf.business-operations.processing-settings.agent-settings.http.client.port*
    - *ipf.business-operations.processing-settings.agent-settings.http.client.deletion-requires-approval*
    - *ipf.business-operations.processing-settings.agent-settings.http.client.creation-requires-approval*
    - *ipf.business-operations.processing-settings.agent-settings.http.client.update-requires-approval*
    - *ipf.business-operations.processing-settings.agent-settings.http.client.call-timeout*
    - *ipf.business-operations.processing-settings.agent-settings.http.client.resiliency-settings*
    - *ipf.business-operations.processing-settings.agent-settlement-settings.http.client.host*
    - *ipf.business-operations.processing-settings.agent-settlement-settings.http.client.port*
    - *ipf.business-operations.processing-settings.agent-settlement-settings.http.client.deletion-requires-approval*
    - *ipf.business-operations.processing-settings.agent-settlement-settings.http.client.creation-requires-approval*
    - *ipf.business-operations.processing-settings.agent-settlement-settings.http.client.update-requires-approval*
    - *ipf.business-operations.processing-settings.agent-settlement-settings.http.client.call-timeout*
    - *ipf.business-operations.processing-settings.agent-settlement-settings.http.client.resiliency-settings*
    - *ipf.business-operations.processing-settings.agent-clearing-settings.http.client.host*
    - *ipf.business-operations.processing-settings.agent-clearing-settings.http.client.port*
    - *ipf.business-operations.processing-settings.agent-clearing-settings.http.client.deletion-requires-approval*
    - *ipf.business-operations.processing-settings.agent-clearing-settings.http.client.creation-requires-approval*
    - *ipf.business-operations.processing-settings.agent-clearing-settings.http.client.update-requires-approval*
    - *ipf.business-operations.processing-settings.agent-clearing-settings.http.client.call-timeout*
    - *ipf.business-operations.processing-settings.agent-clearing-settings.http.client.resiliency-settings*
    - *ipf.business-operations.processing-settings.generic-processing-settings.http.client.host*
    - *ipf.business-operations.processing-settings.generic-processing-settings.http.client.port*
    - *ipf.business-operations.processing-settings.generic-processing-settings.http.client.deletion-requires-approval*
    - *ipf.business-operations.processing-settings.generic-processing-settings.http.client.creation-requires-approval*
    - *ipf.business-operations.processing-settings.generic-processing-settings.http.client.update-requires-approval*
    - *ipf.business-operations.processing-settings.generic-processing-settings.http.client.call-timeout*
    - *ipf.business-operations.processing-settings.generic-processing-settings.http.client.resiliency-settings*
  - HTTP client host and port transport configuration removed for csm-agent and csm-agent-currency
  - Added a time-zones.conf and country-codes.conf that is common for all modules but can be overridden per module
  - Processing entity account type, processing entity account sub-types and agent-identifier types added to show in the drop-down choices in the UI module:

        - *ipf.business-operations.processing-settings.processing-entity-account-types*
        - *ipf.business-operations.processing-settings.processing-entity-account-subtypes*
        - *ipf.business-operations.processing-settings.agent-identifier-types*
    - Added new permissions roles for processing settings that are shortened. **R** meaning read, **C** create, **U** update, **D** delete and **A** approve. **AS** stand for agent settings, **GP** for generic processing settings, **PE** for processing entity and **BF** for bank filtering: *ROLE_DPS_AS_R*, *ROLE_DPS_AS_C*, *ROLE_DPS_AS_A*, *ROLE_DPS_AS_U*, *ROLE_DPS_AS_D*, *ROLE_DPS_GP_R*, *ROLE_DPS_GP_C*, *ROLE_DPS_GP_A*, *ROLE_DPS_GP_U*, *ROLE_DPS_GP_D*, *ROLE_DPS_PE_R*, *ROLE_DPS_PE_C*, *ROLE_DPS_PE_A*, *ROLE_DPS_PE_D*, *ROLE_DPS_PE_U*, *ROLE_DPS_BF_R*, *ROLE_DPS_BF_C*, *ROLE_DPS_BF_A*, *ROLE_DPS_BF_D*, *ROLE_DPS_BF_U*
    - Value path is now conditional if you have more than one possible option in the summary layout config for ODS search in the UI. It can now be a string or an array of strings with the lower the index the more preferable value (if available).

Changed

- **Bug fixes and improvements:**
    - Can no longer create a bank filtering rule with a csm agent that has not been approved
    - Resolved an issue with time zones being misaligned in the pipeline causing test failures
    - Removal of the NCC field for bank filtering
    - Can correctly search by id on csm agent and currency
    - Can correctly search by status in agent settings
    - Run swagger generate and make sure all fields still match new swagger
    - Custom Participant Limits should not be sent on edit in agent settlement if no information has been supplied
    - Dates to be displayed as "Dates" not "DateTimes"
    - Optimise use of screen space on the summary page
    - Make Amount Types in Search results configurable
        - These are now handled by translations rather than static values
    - Fixed issues with viewing payloads in ODS Search
    - Correctly map the system event title
    - Response Code should be toggelable on the UI for display on the Flo Graph

Removed

- Removed Processing Entity Module from Processing Settings
- Removed authorities from JWT token

SEPA CT CSM - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-2-0/release-IPF-2024-2-0-sepa.html

SEPA CT CSM - Changes & Fixes

This page covers the SEPA CT CSM changes and fixes provided in release IPF-2024.2.0.

SEPA CT CSM

> Breaking change - Since the locations for bulker and debulker files and archived files has been changed this may impact any file paths you may have changed from the default.
>
> These will need to be moved to the correct locations in the sepa configurations for ipf.bulker and ipf.debulker. For more information on this change please see IPF-2024.2.0 Bulker & Debulker - Changes & Fixes

New

- File count limits per LAC window introduced, configurable by `ipf.csm.sepa-ct.lac.file-limit-per-lac` By default, produced bulk files are limited to 500 per LAC window
- External outbound inquiry connectors to accept Pacs.027 and Pacs.087 Inquiry messages in XML format.
- Messages from scheme that fail validations will be sent to the client via the sepact-notifications-api
- pacs.008 and pacs.004 from scheme now checked to see if they are duplicates
- Postal addresses of Creditor and Debtor in pacs.008 from the scheme checked to make sure they are either structured of unstructured, not both.

Changed

- Default output location for IQF files is now /tmp/iqf-files.
- Dev Docs updated to include âHow toâ guide for adding code to the Extension points (âFrom Schemeâ and âTo Schemeâ)
- Inbound processing is now a single flow (flow and domain events have been renamed)
- CGS-EOD cutoff time extension now specified by property `ipf.csm.sepa-ct.lac.opening-extension.extension-in-minutes`

Fixed

- NullPointerException when failing to emit MDS during processing files from the scheme
- Transaction processing complete notification sent based on correlations for outbound flows (instead of components)

Migration Steps for IPF-2024.2.0

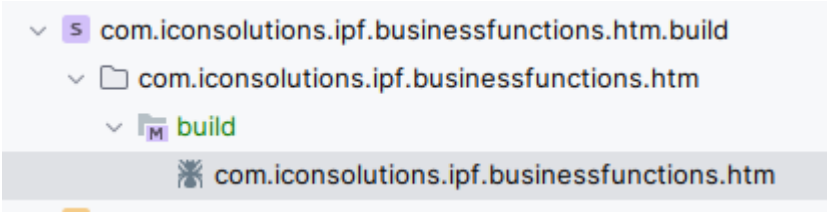
Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-2-0/release-IPF-2024-2-0-migration.html

Migration Steps for IPF-2024.2.0

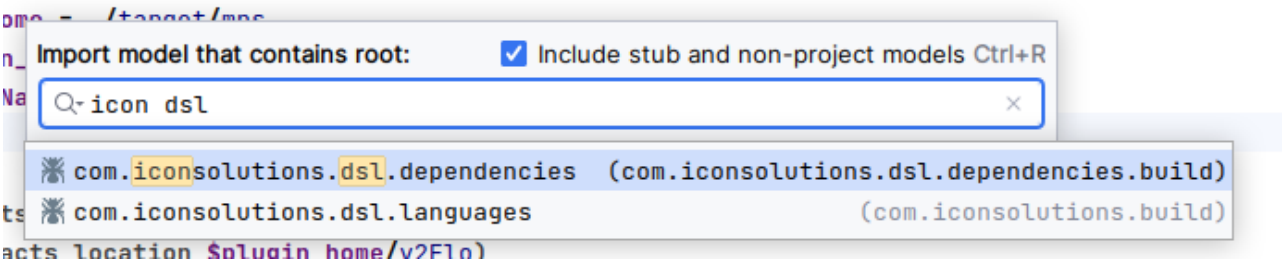
Migration Steps

MPS

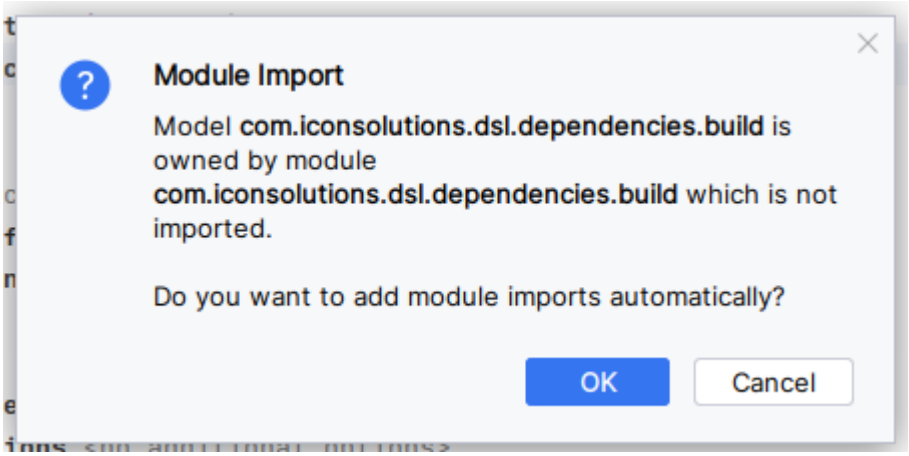
All projects using MPS buildscripts will need to be updated. To do this, open the build script in MPS by clicking on the appropriate file in the navigator:



Then press `Ctrl+R` and ensure the 'include stub and non-project models' selected. Then enter 'icon dsl' in the drop down and select 'com.iconsolutions.dsl.dependencies'



When prompted accept the import of the module.



Repeat the process for 'com.iconsolutions.dsl.languages.

Next in the dependencies section add a new line (press `ENTER`) and then select from the 'com.iconsolutions.dsl.dependencies' from the type ahead (press `Ctrl+Space`).

```
dependencies:
  mps (artifacts location $mps_home)
  v2Flo (artifacts location $plugin_home/v2Flo)
  com.iconsolutions.dsl.dependencies (artifacts location <no artifacts>)
```

In the location section, use the type ahead (press `Ctrl+Space`) to select '$plugin_home', then press `/` and then again select 'com.iconsolutions.dsl.dependencies' from the type ahead.

```
dependencies:
  mps (artifacts location $mps_home)
  v2Flo (artifacts location $plugin_home/v2Flo)
  com.iconsolutions.dsl.dependencies (artifacts location $plugin_home/com.iconsolutions.dsl.dependencies)
```

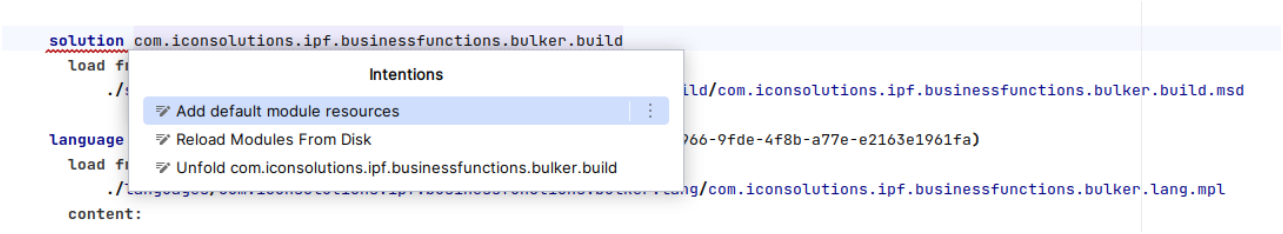
Repeat the process for 'com.iconsolutions.dsl.languages'. Then your dependencies should look like:

```
dependencies:
  mps (artifacts location $mps_home)
  v2Flo (artifacts location $plugin_home/v2Flo)
  com.iconsolutions.dsl.dependencies (artifacts location $plugin_home/com.iconsolutions.dsl.dependencies)
  com.iconsolutions.dsl.languages (artifacts location $plugin_home/com.iconsolutions.dsl.languages)
```

Now, you should now see an error on your solution section for the build:

```
solution com.iconsolutions.ipf.businessfunctions.bulker.build
  load from
      ./solutions/com.iconsolutions.ipf.businessfunctions.bulker.build/com.iconsolutions.ipf.businessfunctions.bulker.build.msd
```

To resolve, click on the underlined text and press `Alt+Enter` to bring up the intentions popup and select 'Reload Modules From Disk'.



That should resolve the error. Finally, you need to regenerate the build script. To do this, simply right click on the build solution in the navigator and select 'Rebuild Solution'.

You should now be able to build your project through maven as normal.

Update pacs.003 version

The supported pacs.003 message type within the IPF ISO20022 Model has been upgraded from pacs.003.001.07 to pacs.003.001.08.

If your IPF solution includes pacs.003 messages, you will need to make the following type changes:

Version 7 Type	Version 8 Type
com.iconsolutions.iso20022.datatypes.AddressType2Code	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.direct_debit_mandate.amendment_information_details11.AmendmentInformationDetails11	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.organisation.branch_and_financial_institution_identification5.BranchAndFinancialInstitutionIdentification5	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.organisation_identification.branch_data2.BranchData2	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.cash_account.cash_account24.CashAccount24	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.charges.charges2.Charges2	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.person_identification.contact_details2.ContactDetails2	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.person.date_and_place_of_birth.DateAndPlaceOfBirth	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.date_time_period.date_period_details.DatePeriodDetails	com.iconsolutions.iso2002

Version 7 Type	Version 8 Type
com.iconsolutions.iso20022.message.components.direct_debit.direct_debit_transaction9.DirectDebitTransaction9	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.direct_debit.direct_debit_transaction_information21.DirectDebitTransactionInformation21	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.organisation_identification.financial_institution_identification8.FinancialInstitutionIdentification8	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.definitions.payments_clearing_and_settlement.pacs003.FIToFICustomerDirectDebitV07	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.garnishment.garnishment1.Garnishment1	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.payment.group_header50.GroupHeader50	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.direct_debit_mandate.mandate_related_information11.MandateRelatedInformation11	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.party_identification_information.name_and_address10.NameAndAddress10	com.iconsolutions.iso2002
com.iconsolutions.iso20022.datatypes.NamePrefix1Code	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.organisation_identification.organisation_identification8.OrganisationIdentification8	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.party_identification_information.party11_choice.Party11Choice	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.party_identification_information.party_identification43.PartyIdentification43	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.payment_identification.payment_identification3.PaymentIdentification3	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.payment_processing.payment_type_information25.PaymentTypeInformation25	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.person_identification.person_identification5.PersonIdentification5	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.postal_address.postal_address6.PostalAddress6	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.document.remittance_information11.RemittanceInformation11	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.contact_point.remittance_location4.RemittanceLocation4	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.contact_point.remittance_location_details1.RemittanceLocationDetails1	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.cash_settlement.settlement_instruction2.SettlementInstruction2	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.document.structured_remittance_information13.StructuredRemittanceInformation13	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.tax.tax_amount1.TaxAmount1	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.tax.tax_information4.TaxInformation4	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.tax_period.tax_period1.TaxPeriod1	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.tax_record.tax_record1.TaxRecord1	com.iconsolutions.iso2002
com.iconsolutions.iso20022.message.components.tax.tax_record_details1.TaxRecordDetails1	com.iconsolutions.iso2002

> When comparing the pacs.003.001.07 and pacs.008.001.08 XSDs, you will notice that a small number of fields have been added or removed. As a result, you may need to make changes to your code beyond the type changes above. For example, if you previously populated a field that is now removed.

Migrating to `EVENT_STREAM_PER_FLOW` streams

To support deploying new flow versions via rolling upgrades of your orchestration services, all the journal processors defined by the service have to be switched over to use `EVENT_STREAM_PER_FLOW` as their `event-streaming-type`.

Before you can perform the configuration switch, however, you will need to scale down your service deployment to zero and run a migration similar to the one specified below.

```
 // all the unique flows in your service, as indicated by the
// `ipf.behaviour.event-processor.flows` configuration property
let flows = [
    "moduleA.FlowAV1",
    "moduleA.FlowAV2",
    "moduleB.FlowBV5",
    "moduleC.FlowBV10"
]

db.getCollection("mongoOffset")
    .find({"_id.tag": /tag-.+/})
    .forEach(doc => {
        flows.forEach(flow => {
            let flowTag = JSON.parse(JSON.stringify(doc));
            flowTag.offset.objectId = doc.offset.objectId;
            flowTag._id.tag = doc._id.tag.replace("tag", flow);

            try {
                db.getCollection("mongoOffset").insertOne(flowTag);
            } catch (e) {
                console.log(e)
            }
        });
    });
```

> It is imperative that you list all the flows present in the current version of your orchestration service. Any flow that is missed will have all of its events reprocessed by the journal processor, which may cause duplicate notifications to be sent to the customers, incorrect metrics to appear on the dashboards or just cause a massive slowdown in processing of new domain events since the processor will be starting from the beginning of the journal.

To get the correct list of active flows, you can run the following curl command against an instance of your orchestration service:

```
 curl your_service_host:port/actuator/info | jq | grep "ipf.behaviour.event-processor.flows"
```

Alternatively, you can run the following test snippet from within the module that holds your service's `SpringBootApplication` — the output will be the migration script that you need to execute against your database.

```
 import com.typesafe.config.ConfigFactory;
import org.junit.jupiter.api.Test;
import java.util.stream.Collectors;

public class MigrationGenerator {

    @Test
    void generateMigration() {
        var flowsAsJsArray = ConfigFactory.parseResources("ipf-impl.conf")
                .withFallback(ConfigFactory.parseResources("ipf.conf"))
                .withFallback(ConfigFactory.defaultReference())
                .resolve()
                .getStringList("ipf.behaviour.event-processor.flows")
                .stream().distinct()
                .map("\"%s\""::formatted)
                .collect(Collectors.joining(",\n ", "let flows = [\n ", "\n]\n"));

        var migrationScriptWithoutFlows = """
                db.getCollection("mongoOffset")
                    .find({"_id.tag": /tag-.+/})
                    .forEach(doc => {
                        flows.forEach(flow => {
                            let flowTag = JSON.parse(JSON.stringify(doc));
                            flowTag.offset.objectId = doc.offset.objectId;
                            flowTag._id.tag = doc._id.tag.replace("tag", flow);
                            try {
                                db.getCollection("mongoOffset").insertOne(flowTag);
                            } catch (e) {
                                console.log(e)
                            }
                        });
                    });
```

```
                """;

        System.out.println(flowsAsJsArray);
        System.out.println(migrationScriptWithoutFlows);
    }
}
```

Migrating to ODS Inquiry API V2

- Versions details and the complete V2 ODS Inquiry API migration guide can be found on the ODS API versioning overview page

> It is important to note that a config change **MUST** be applied if you are already using the V1 client connectors and do not want to immediately migrate to the V2 client connectors. Your configuration must be updated to override the client connector version configuration as follows: `ods.inquiry.client.version=1` This will ensure your current V1 client connectors remain enabled and a code change will not be required. See V1 Inquiry API client documentation for more information on the V1 client APIs that can be enabled.

Migrating to the latest test framework

> There have been minor changes to the package structures in the test framework. 'ipf-test-fw-core' has now moved to 'ipf-test-fw-whitebox'.

> The test-fw-extensions-xxx have now been removed from the core package to ensure that unrequired extensions are not always loaded. As a consequence, test implementations that relied on these dependencies will now have to explicitly declare them. These are listed below and are usually required when using one of the provided message transporters:
>
> - test-fw-extensions-http
> - test-fw-extensions-kafka
> - test-fw-extensions-jms

Jackson serialisation behaviour change

Please note that as a result of updating Jackson to 2.17.x (by updating Spring Boot 3.3.x), Jackson is now no longer serialising objects whose members are all null. For example, in IPF 2024.1 and older an object that looks like this:

```
{
   "myObject": {
       "someKey": "someValue",
       "someBlankParent": {}
   }
}
```

Will now be serialised (and deserialised) like this:

```
{
   "myObject": {
       "someKey": "someValue"
   }
}
```

(i.e. `someBlankParent` is now totally absent instead of being an empty shell)

This could present itself as a `NullPointerException` where null checks were assuming that there was an empty shell as a parent object, when one no longer exists. The solution is to add a null check to the parent object as well.

Please note that no data is being lost since the objects not being serialised were empty to begin with. Any parent object containing at least one value will be serialised along with its data-bearing members.

TransactionCacheEntry collection - change in index names

There was a change in index names that need to be created for transactionCacheEntry mongodb collection (if they are not created and index creation is enabled).

- findByTypeAndHashIndex is renamed to hash_1_type_1
- findByTypeAndHashAndMessageIdIndex is renamed to hash_1_type_1_messageId_1.

If your mongodb already has those indexes with old names, you will get this error in the log:

```
 TransactionCacheIndexInitialiser.accept - Failed to ensure index on TransactionCacheEntry
org.springframework.data.mongodb.UncategorizedMongoDbException: Command failed with error 85 (IndexOptionsConflict): 'Index
```

```
already exists with a different name: findByTypeAndHashIndex' on server
```

If you want to fix this error, rename existing indexes to new names.

Release Notes for IPF-2024.2.1

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-2-1/release-IPF-2024-2-1.html

```
```

Release Notes for IPF-2024.2.1

This page details everything required to get going on IPF Release 2024.2.1 made available on **08th October 2024**.

Change Spotlight

- Archetype has now been replaced by **ipf-project-scaffolder**

The latest version of the scoffolder is **1.0.3.1**

Instructions on how to use the scaffolder to create new projects are here

Fix Spotlight

- Fixed the response codes when using the TIPS CSM Scheme pack (*PAY-11441*)
- Additional changes to CSM Reachability (*PAY-11589*)

Release Details

Binaries

The core binaries are available from IPF-Releases.

Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Akka/Alpakka binaries from 2.9.x onwards are no longer on Maven Central and are available from the Akka Library Repository which Icon mirror.

To enable the new mirror, add the following to your settings.xml in the `<repositories>` section:

```
 <repository>
  <id>akka-repository</id>
  <name>Akka library repository</name>
  <url>https://nexus.ipf.iconsolutions.com/repository/akka-repository</url>
</repository>
```

Some MPS dependencies are also mirrored by Icon and can be found by adding the following repository to the same section:

```
 <repository>
  <id>icon-thirdparty</id>
  <name>IPF icon-thirdparty repository</name>
  <url>https://nexus.ipf.iconsolutions.com/repository/icon-thirdparty</url>
</repository>
```

Add the following to your `<servers>` section for each new repository:

```
 <server>
  <id>Reposity_name_here</id>
  <username>xxxx</username>
  <password>xxxx</password>
</server>
```

Where `xxxx` is the username and password repectively provided to you by Icon.

Developer App

The latest version of the Developer App is **2.3.8**

2024.2.1 Jar/Pom list

The Jars and Poms for 2024.2 and the associated versions are listed here: 2024-2-artifacts.xlsx

---

Release Notes for IPF-2024.1.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-1-0/release-IPF-2024-1-0.html

Release Notes for IPF-2024.1.0

This page details everything required to get going on IPF Release 2024.1.0 made available on 17 May 2024.

Change Spotlight

> - Java 17 is now the minimum version required to build and develop IPF
> - New version of MPS is 2022.3.1 and you will need to apply the steps found in the attached Migration Steps

- New versions of Spring and Spring Boot: 6.1.x and 3.2.x respectively
- New version of JBehave: 5.2.0
- Changes and fixes to the following Modules: Bank Filtering, Working Day Service, Notification Service and the Scheme Packs
- V2 of the CSM Reachability APIs, added dynamic configuration and changed data structures
- Added processing of new camt message types and ODS integration to SEPA CT CSM
- Added resiliency, timeouts and ODS integration for Bulker/Debulker
- New features and fixes for ODS and Operational Dashboard/GUI

Fix Spotlight

- TIPS Scheme pack now maps the BICs from the Canonical payload.content.GrpHdr.InstgAgt.FinInstnId.BICFI & payload.content.GrpHdr.InstdAgt.FinInstnId.BICFI as received - (*PAY-10258*)
- Validation to ensure that terminal events are indeed terminal - (*PAY-10250*)
- Improved Unit of Work ID generation in RT1 to ensure uniqueness with error handling for edge cases - (*PAY-9851*)
- Alternative ID value in ODS Summary response now returns the actual value, and does not return the value as lower case - (*PAY-9830*)
- Fixed JmsTestTransporter to add user defined request message types as supported messages - (*PAY-9311*)

Release Details

Binaries

The core binaries are available from IPF-Releases.

Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Akka/Alpakka binaries from 2.9.x onwards are no longer on Maven Central and are available from the Akka Library Repository:

```
<repository>
  <id>akka-repository</id>
  <name>Akka library repository</name>
  <url>https://repo.akka.io/maven</url>
</repository>
```

Archetype

The latest version of the Archetype is **5.0.9**

Developer App

The latest version of the Developer App is **2.1.2**

2024.1 Jar/Pom list

The Jars and Poms for 2024.1 and the associated versions are listed here: 2024-1-artifacts.xlsx

Migration Steps

Java 17

As was announced in the previous two releases, release 2024.1.0 makes Java 17 mandatory for developing, building, and running IPF. Please ensure that you are using a Java 17 JDK in:

- Your development environment
- Your CI environment
- Your container base image

MPS 2022.3

We have now upgraded to using MPS 2022.3 for the design environment which allows us to introduce new features to our Flo Lang DSL. Please ensure you have installed MPS 2022.3 prior to using IPF 2024.1:

- Windows: download.jetbrains.com/mps/2022.3/MPS-2022.3.1.exe
- Linux: download.jetbrains.com/mps/2022.3/MPS-2022.3.1.tar.gz

You will also need the corresponding `2022.3.24237.f0c4f91` version of `com.mbeddr:platform`, which is located in the Itemis MPS Repository.

> - MPS project generation in windows is not correctly moving files
>   - If projects are built on CI, and the CI system is unix-based there's nothing to change
> - This has been fixed for the next release. As a workaround in the meantime, add the contents c `[myapp]-domain/domain/target/ipf-conf-merge-files/merged-ipf-conf-files/conf/[name of model]/ipf.conf` to any other config file that isn't part of the autogenerated code e.g. ir `ipf-impl.conf` of the `-app` module

> Migration Steps

Optional Modules - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-1-0/release-IPF-2024-1-0-aom.html

Optional Modules - Changes & Fixes

This page covers the optional module changes and fixes provided in release IPF-2024.1.0.

Bank Filtering

- Fixed responses for getAll BankFilteringRules, CsmAgents, ProcessingEntity in Swagger UI.
- Filtering by currency was taking lowest severity, now it takes the highest.
- The location of country code in BIC is fixed.
- Removed filtering by NCC, this will be re-introduced in the PI 2024.2
- Rule direction changed from DEBTOR, CREDITOR, BOTH to DEBTOR, CREDITOR, ANY. Handling direction ANY implemented to work as logical OR, which means it's applied to input debtor agent or creditor agent.
- Country code extraction from input BIC handles successfully any BIC (even 4 characters BIC).

Working Day Service

- Working-days-service separated to api, components and app repository: working-days-service, working-days-service-api and working-days-service-app.

Scheme Packs

- In memory configuration holder which has a standard format defined for a dynamic configuration.
- Services RT1, TIPS, SIC5, and STEP2 SCT are updated to use DynamicConfigHolder, as the first step in updating CSM services to use dynamic configuration. DynamicConfigHolder is filled with data from HOCON configuration.

Notification Service

- IPF Core Notification Service provides a capability to send notification messages to different Kafka topics, based on a processing entity attribute enabled.
- Added optional predicates to PaymentStatusNotification endpoints that can be configured through `payment-status-notification.notification-settings.endpoints`:

```
[
  {
    "topic": "PAYMENT_STATUS_NOTIFICATION",
    "predicate": "eventId == 1"
  },
  {
    "topic": "PAYMENT_SECONDARY_TOPIC",
    "predicate": "eventId == 2"
  }
]
```

Bulker & Debulker - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-1-0/release-IPF-2024-1-0-bulker.html

Bulker & Debulker - Changes & Fixes

This page covers the Bulker & Debulker module changes and fixes provided in release IPF-2024.1.0.

Bulker

> In the new release, Bulker now passivates bulks after 2 minutes by default. It has been identified that there are some issues when rehydrating actors after passivation has occurred. This will be fixed in the next release. To mitigate this in the current release, it is recommended to increase the bulker passivation timeout to a larger value than what you would expect the bulk to close. For Example, if the largest bulk window is 4 hours, it is recommended to set the value below to at least this value or even higher in your application.conf file: `ipf.bulker.aggregate.actor-idle-timeout = 5h`

New

- Integration with ODS
- Automatic finalisation due to âByte Fullnessâ
- Support for client to component id when sending AddComponentCommand, enforcing idempotency in ipf-bulker-aggregate-akka
- Enhanced resilience and error reporting
- Current bulk size is returned in AddComponent response
- The following properties have been added to support command retries:
    - `ipf.bulker.aggregate.retry.processing-timeout`
    - `ipf.bulker.aggregate.retry.max-attempts`
    - `ipf.bulker.aggregate.retry.backoff-factor`
    - `ipf.bulker.aggregate.retry.jitter-factor`
- Other added config:
    - `ipf.bulker.aggregate.actor-idle-timeout = 2m` (idle duration after which actor will be passivated)
    - `maximum-component-size` (optional, maximum number of components allowed before bulk is auto closed, configured at the bulk level)

Changed

- The single bulk behaviour has now been converted to a regular Akka Actor, components saved directly to the ComponentStore as commands are received.
- The recurring bulk behaviour has now been converted to a durable state behaviour
- Capacity limits applied âper Bulkâ not âper Instanceâ.
- File production recovery on system failure
- Improved Documentation
- Handle recurring bulk commands across multiple nodes
- Preventing creation of two single bulk instances by recurring bulk if command to create a new single bulk instance is received while new single bulk instance is being created.
- The following properties have been changed `maximum-bulk-size` (optional, maximum allowed size for bulk, configured at the bulk level)

Removed

- The following property has been removed `ipf.bulker.auto-close.maximum-components`

Debulker

New

- Integration with ODS
- System events `DuplicateCheckFailed` and `SchemaValidationFailed`

- Added global statuses to debulker flow

Changed

- Splitters to use flux instead of flow.publisher
- Improved Documentation
- Enhanced resilience and error reporting
- Fixed conflicting config paths

CSM Reachability - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-1-0/release-IPF-2024-1-0-csm-reach.html

CSM Reachability - Changes & Fixes

This page covers the CSM Reachability V2 improvements, changes and fixes provided in release IPF-2024.1.0.

CSM Reachability and Industry Data Ingestion

CSM Reachability API

Added

- API versioning introduced and new version of API /v2
- Existing endpoints defined under version v2
    - `/v2/bicvalidation`
    - `/v2/select-csm-agent`
    - `/v2/validate-csm-reachability`
    - `/v2/iban/deconstruct`
    - `/v2/party-entities`
- New Domain API endpoints
    - `/v2/settlement-agents`
    - `/v2/processing-entities/{processingEntity}`
- OpenApi specification csm-reachability-service-api-v2.yaml.
- API model, generated in packages dto.v1 or dto.v2, used in controllers and client port implementations
- Service model, generated in dto package, used in service implementation
- API connector created for:
    - `/v2/iban/deconstruct`
    - `/v2/validate-csm-reachability`
    - `/v2/bicvalidation`
    - `/v2/select-csm-agent`
    - `/v2/settlement-agents`
    - `/v2/party-entities`
    - `/v2/processing-entities/{processingEntity}`

Changed

- OpenAPI specification `csm-reachability-service-api.yaml` replaced with `/static/csm-reachability-service-api-v2.yaml`

Removed

- `/bicvalidation`
- `/select-csm-agent`
- `/validate-csm-reachability`
- `/iban/deconstruct`
- `/party-entities`

CSM Reachability

New

- Dynamic Configurations:
    - New dynamic configurations added (supported by DPS V1 APIs)

        - AgentSettings
        - AgentClearingSettings
        - CsAgentSelectionSettings
        - GenericProcessingSettings
            - IntraEntityParties
            - ProcessingEntityAccounts
            - SupportedTransferCurencies
        - New searchable fields for ProcessingEntity
        - Settings are moved from `dynamic-settings-workflow` to `csm-reachability`.
        - ISO 3166-2 country codes are configurable with `property ipf.csm-reachability.setting.country-codes`
        - ISO 4217 currency codes are configurable with `property ipf.csm-reachability.setting.currency-codes`
- Service Implementations:
    - Common module for EntityIdentifiers and EntityAddresses(which has ConfigurableEnum)
    - `ipf.conf` file with swagger configuration in the csm-reachability-setting-management module
    - Select CSM Agent implementation to work with new data model and with updated input and output definitions (/v2 version of the API)
    - Validate CSM Reachability implementation using new data structures
    - ConfigurableEnum annotation for client defined enum values

Changed

- Dynamic Configurations:
    - CsmAgent data structure (replaced by AgentSettings)
    - CsmAgentCurrency data structure (replaced by new structures: AgentClearingSettings and AgentSettlementSetting)
        - OpeningTime and ClosingTime are now LocalDate type, not String, TimeZone field mandatory when opening or closing time provided
    - CsmAgentSelectionOrder changed into CsAgentSelectionSettings
    - ProcessingEntity data structure re-designed, some fields are removed, including BIC which is now combination of entityIdentifierType and entityIdentifierValue:
        - added :
            - field 'entityIdentifier' - list of Entityidentifiers
            - field 'entityAddress'
        - removed :
            - field 'clearingSystemIdentification'
            - field 'debtorAgentName'
            - field 'onUsCurrencyList' - list of OnUsCurrency
            - field 'suspenseAccountList' - list of SuspenseAccounts
        - changed:
            - field 'bic' is now combination of 'entityIdentifierType' and 'entityIdentifierValue' field 'postalAddressStreetName' now is a part of Address property as 'streetName'
            - field 'postalAddressBuildingName' now is a part of Address property as 'buildingName'
            - field 'postalAddressPostalCode' now is a part of Address property as 'postCode'
            - field 'postalAddressTownName' now is a part of Address property as 'townName'
            - field 'postalAddressCountryCode' now is a part of Address property as configurable 'country'

- Added LedgerCurrency attribute to Processing Entity configuration.

Database Impact

- Settings that are no longer used and collections that need to be dropped are:
  - CsmAgent (`settings-csm-agent`)
  - CsmAgentSelectionOrder (`settings-csm-agent-selection-order`)
  - CsmAgentCurrency (`settings-csm-agent-currency`)
- New settings should be used, and new collections created:
  - AgentSettings (`settings-agent-settings`)
  - AgentClearingSettings (`settings-agent-clearing-settings`)
  - AgentSettlementSettings (`settings-agent-settlement-settings`)
  - CsAgentSelectionSettings (`settings-cs-agent-selection-settings`)
  - GenericProcessingSettings (`settings-generic-processing-settings`)

CSM Reachability Runnable Application

- Swagger UI page configured to be at path `/swagger-ui/index.html`

Data Ingestion

New

- Data ingestion implementation moved to new repository csm-reachability-data-ingestion
- Six Bank Master load to Party Entity directory is now supported. Json files are parsed and loaded to party-entity collection.
- One new module party-entity with 2 submodules: party-entity-swift and party-entity-six
- New system event FileEntrySkippedEvent
- Fixed file ingestion throttling.

Changed

- Previous config for party entity file ingestion was `partyentity.process-bank-directory-plus.enabled = true`, now is changed to `swift.bankplus.process-bank-directory-plus.enabled = true`
- The old implementation has been kept for backward compatibility. Also, since now we have support for json file ingestion in party entity new configuration is added in new module for bank master `six.bankmaster process-bank-master.enabled = true`. For example:

```
party-entity.swift.bankplus {
  process-bank-directory-plus.enabled = true
  file-ingestion-connector.file-ingestion {
    files-directory = "/static/files"
    directory-id = "party-entity-swift"
    initial-delay = 9s
    interval = 1h
  }
}

six.bankmaster {
  process-bank-master.enabled = true
  file-ingestion-bankmaster-connector {
    file-ingestion {
      files-directory = "/static/files"
      directory-id = "party-entity-bank-master"
      initial-delay = 9s
      interval = 1h
    }
  }
}
```

ODS & GUI - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-1-0/release-IPF-2024-1-0-gui-ods.html

ODS & GUI - Changes & Fixes

This page covers the Operation Data Store (ODS) changes and fixes provided in release IPF-2024.1.0.

ODS

New

- IPF Metrics Processor
    - The IPF Metrics Processor is an optional add-on. It utilises kafka to produce payment metrics, which includes counts and durations of finished payments, and counts of error codes.
    - Additonal documentation and implementation guidelines found here. (Currently only staging docs, Will update link to external docs upon doc release)
- Camt027 and Camt087 MDS Object Support
    - Camt.027 and camt.087 MDS Data can now be sent via the IPF Processing Data Kafka topic and ingested by ODS.
    - There are two new ODS Inquiry API endpoints to query for these additional MDS objects:
        - `/catalogue/mds-objects/CAMT_027`
        - `/catalogue/mds-objects/CAMT_087`
    - Both MDS objects can also be returned as part of the following ODS Inquiry API endpoints responses:
        - `/all/mds-objects/{odsObjectId}`
        - `/all/mds-objects/{mdsObjectId}/history`
        - `/views/details/{unitOfWorkId}`

GUI

Changed

- Developer app updates and improvements
    - Added summary, MDS and PDS tabs and improved the rendering of the graphs in the developer app
- New role to only allow the export functionality for specific users
- Implemented ngRx's linting rules as standard for our projects for a better standard of the codebase
- Updating the node and npm versions across our projects as our previous versions were now at end of life. Currently at node 20 and npm 10
- Split the backend config endpoint per module so any module that required a config no longer required ods-payment-search
- Updated the existing multiselect component to include a select all option
- Some general styling around the input field and requirements
- New navigation and route redesign
- New global navigation redesign and implementation for a more consistent feel and better user experience.

Fixed

- Fixed an issue with the PDS history not showing any data
- Fixed an issue with sonar gate failing on ops-gui-framework and not picking up all the code correctly
- Removed `CLIENT_REQUEST_ID` from alternative Ids on the summary card
- Bank Filtering fixes
    - CSM Agent ID is now optional
    - Direction field updated based on service changes
- The GUI now displays the time zone offset against UTC for date time fields

- Creditor and Debtor identifier value fields are now correctly names
- Fixed an issue with swagger docs not matching the expected response for processing entity endpoint

GUI Documentation & Tutorial

- Updating docs and details about our modules
    - Version info
    - Metric
    - Processing entity mechanism and Permissions lib
    - Cluster health

SEPA CT CSM - Changes & Fixes

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-1-0/release-IPF-2024-1-0-sepa.html

SEPA CT CSM - Changes & Fixes

This page covers the SEPA CT CSM changes and fixes provided in release IPF-2024.1.0.

SEPA CT CSM

- New
    - Complete camt.027 processing
    - Complete camt.087 processing
    - Complete pacs.028 processing
    - EOD generation of pacs.002 when no response from scheme
    - Identification and highlighting of Group Header errors in CVF response
    - Integration with DeadletterAppenders
    - Extension points for Inbound and outbound files
    - Camt.056 PCF Processing
    - Integration with ODS
    - Properties added under `ipf.csm.sepa-ct.extension-point` to allow clients to optionally call an extension point during message processing
    - Added `ipf.csm.sepa-ct.processing-entity.pacs008-duplicate-message-check-period` to set duplicate validation period for Pacs.008
- Changed
    - camt.029 processing includes IQF use
    - Enhanced maintainability, resilience and error reporting
    - Configurable EOD trigger time
    - Updating duplicate validator to include idempotency key
    - Improved Documentation
    - Replaced country codes JSON file with hocon config
- Removed
    - Dependency on mps-domain project in business day calculator as its not used

> In the new release, Bulker now passivates bulks after 2 minutes by default. It has been identified that there are some issues when rehydrating actors after passivation has occurred. This will be fixed in the next release. To mitigate this in the current release, it is recommended to increase the bulker passivation timeout to a larger value than what you would expect the bulk to close. For Example, if the largest bulk window is 4 hours, it is recommended to set the value below to at least this value or even higher in your application.conf file: `ipf.bulker.aggregate.actor-idle-timeout = 5h`

Migration Steps for IPF-2024.1.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-1-0/release-IPF-2024-1-0-migration.html

Migration Steps for IPF-2024.1.0

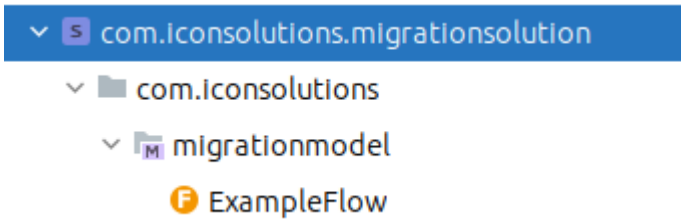
Pre-requisites

The 2024.1.0 release has upgraded to Java 17 and MPS v2022.3.1. You'll need to ensure you have these versions available and installed before upgrading.

Before you start

Tiles preparation

As part of the upgrade, we recommend that you adopt the new tiles approach to DSL configuration. To do this, you'll need to be able to provide the model and solution names for your project. These can be found by looking in your MPS project:



In the screenshot above, the solution name is `com.iconsolutions.migrationsolution` and the model name is `com.iconsolutions.migrationmodel`. Make sure to remember to include a package name if defined (like the `com.iconsolutions` here), and note that case sensitivity is important.

> You can directly copy the solution and model names by right-clicking on them in MPS.

Java 11 references removal

Any hardcoded Java 11 references in the pom need to be removed before starting the migration process, e.g.:

```
<properties>
   <maven.compiler.source>11</maven.compiler.source>
   <maven.compiler.target>11</maven.compiler.target>
   <maven.compiler.release>11</maven.compiler.release>
</properties>
```

Base Docker image update

Any application images now need to be built on top of a JDK 17 base. Any references to a JDK 11 base image (e.g. `openjdk:11`) in docker image generation should be updated to a JDK 17 one (e.g. `ubi8-minimal-openjdk-17:latest`)

Migrate to the new `ipf-bom`

As part of the 2024.1.0 release, we have deployed a new `ipf-bom`, which provides access to all IPF utilities in a single place. We recommend you switch to using this BOM as part of the migration process. For example, if the current parent of your project is `ipf-release-core-bom`, you should now use:

```
<parent>
   <groupId>com.iconsolutions.ipf</groupId>
   <artifactId>ipf-bom</artifactId>
   <version>2024.1.0</version>
</parent>
```

> The new `ipf-bom` standardises the property definitions for cinnamon. Therefore, anything that used to refer to the old property `lightbend-cinnamon.version` should now refer to the new name `cinnamon.version`. Note that if using docker image generation, this includes the generation command within the application pom for example:
>
> ```
>  <cmd>exec java -javaagent:/${project.artifactId}/lib/com.lightbend.cinnamon-cinnamon-agent-${cinnamon.version}.jar -
> agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=*:5005 -cp
> "/${project.artifactId}/conf:/${project.artifactId}/lib/*" $IPF_JAVA_ARGS
> com.iconsolutions.com.iconsolutions.migrationsolution.app.Application -startup</cmd>
> ```

Update the pom configuration for DSL projects

As mentioned above, the structure of the DSL projects has been changed. This provides a number of benefits going forward in regard to simplifying the integration of DSL projects within your application. To migrate, the following steps need to be performed:

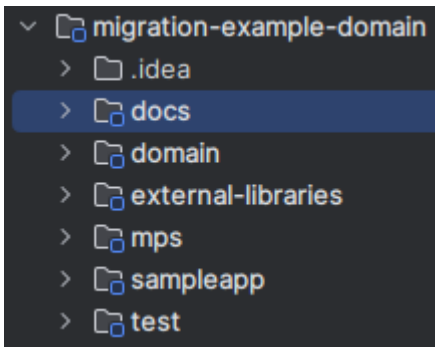
- In your root pom, add the following properties:

```
<properties>
   <solution.name>com.iconsolutions.migrationsolution</solution.name>
   <model.name>com.iconsolutions.migrationmodel</model.name>
</properties>
```

Where the `<solution.name>` and `<model.name>` values are those described in the Before you start section above.

Update the parent of the domain submodules

The parent of the submodules from the `[name of app]-domain` module no longer needs to be a special reference to the respective `flo-starter-xxx` artifacts, and should now just be the 'natural' parent of the module. For example, if your MPS artifacts all live under a module "migration-example-domain" like this:



Then the parent for each of the submodules should now look like this:

```
<parent>
   <groupId>com.iconsolutions.com.iconsolutions.migrationsolution.domain</groupId>
   <artifactId>migration-example-domain</artifactId>
   <version>0.0.1-SNAPSHOT</version>
</parent>
```

After this change, you no longer need the additional `<groupId>…</groupId>` tag outside the parent, and so it can be removed.

> You can confirm the parent has been updated correctly in each of the submodules by comparing them against the parent in the existing `external-libraries` submodule.

Update dependency version references

Any dependency versions in the submodules that refer to the property `flo.version` need to be updated to `icon-flo.version`:

```
<dependency>
   <groupId>com.iconsolutions.ipf.core.flow</groupId>
   <artifactId>library-dependencies</artifactId>
   <version>${icon-flo.version}</version>
</dependency>
```

Also, in the `test` submodule, the `flo-test-common` dependency version should be changed from `project.parent.version` to `icon-flo.version`:

```
<dependency>
   <groupId>com.iconsolutions.ipf.core.flow</groupId>
   <artifactId>flo-test-common</artifactId>
   <version>${icon-flo.version}</version>
   <type>test-jar</type>
   <scope>test</scope>
</dependency>
```

Add the relevant tile(s) to the build section of the pom

Next, in each of the submodule poms, we need to add the relevant tile to tell them what to build. To do this we add the following section to the

pom:

```
<build>
   <plugins>
       <plugin>
           <groupId>io.repaint.maven</groupId>
           <artifactId>tiles-maven-plugin</artifactId>
           <version>${tiles-maven-plugin.version}</version>
           <extensions>true</extensions>
           <configuration>
               <tiles>
                   <tile>com.iconsolutions.ipf.core.flow:flo-tilename-tile:${icon-flo.version}</tile>
               </tiles>
           </configuration>
       </plugin>
   </plugins>
</build>
```

Where we replace the '**tilename**' with the appropriate tilename for each of the submodules as defined in the list below:

- `docs` uses `docs`
- `domain` uses `domain`
- `mps` uses `mps-plugin`
- `test` uses `test`
- `sampleapp` (if applicable) uses `sampleapp`

So, for example, the `<tile>` tag for the `mps` submodule should be:

```
<tile>com.iconsolutions.ipf.core.flow:flo-mps-plugin-tile:${icon-flo.version}</tile>
```

> If your submodule already has a `<build>` section in the pom.xml (such as the `mps` submodule), then the tile will need to be added as an additional plugin in this section.

> The above definitions assume that your project is not going to be used as a dependency of another MPS project. If it is, we need to change th `mps` definition so that is uses two different tiles:
>
> ```
>  <tiles>
>     <tile>com.iconsolutions.ipf.core.flow:flo-mps-tile:${icon-flo.version}</tile>
>     <tile>com.iconsolutions.ipf.core.flow:flo-mps-archive-tile:${icon-flo.version}</tile>
>  </tiles>
> ```
>
> instead of just the one defined above.

Add the required dependencies

Finally, we need to add some additional depedencies to the modules:

The `domain` submodule requires the following two additions:

```
<dependency>
   <groupId>com.iconsolutions.ipf.core.flow</groupId>
   <artifactId>flo-starter-domain</artifactId>
</dependency>
<dependency>
   <groupId>com.iconsolutions.ipf.core.flow</groupId>
   <artifactId>flo-starter-domain-test</artifactId>
   <scope>test</scope>
</dependency>
```

The `mps` submodule needs this dependency added:

```
<dependency>
   <groupId>com.iconsolutions.ipf.core.flow</groupId>
   <artifactId>flo-starter-mps</artifactId>
</dependency>
```

The `sampleapp` submodule needs this dependency added:

```
<dependency>
```

```
    <groupId>com.iconsolutions.ipf.core.flow</groupId>
    <artifactId>flo-starter-sampleapp</artifactId>
</dependency>
```

The `test` submodule needs this dependency added:

```
<dependency>
    <groupId>com.iconsolutions.ipf.core.flow</groupId>
    <artifactId>flo-starter-test</artifactId>
    <scope>test</scope>
</dependency>
```

Migration and Updates within MPS

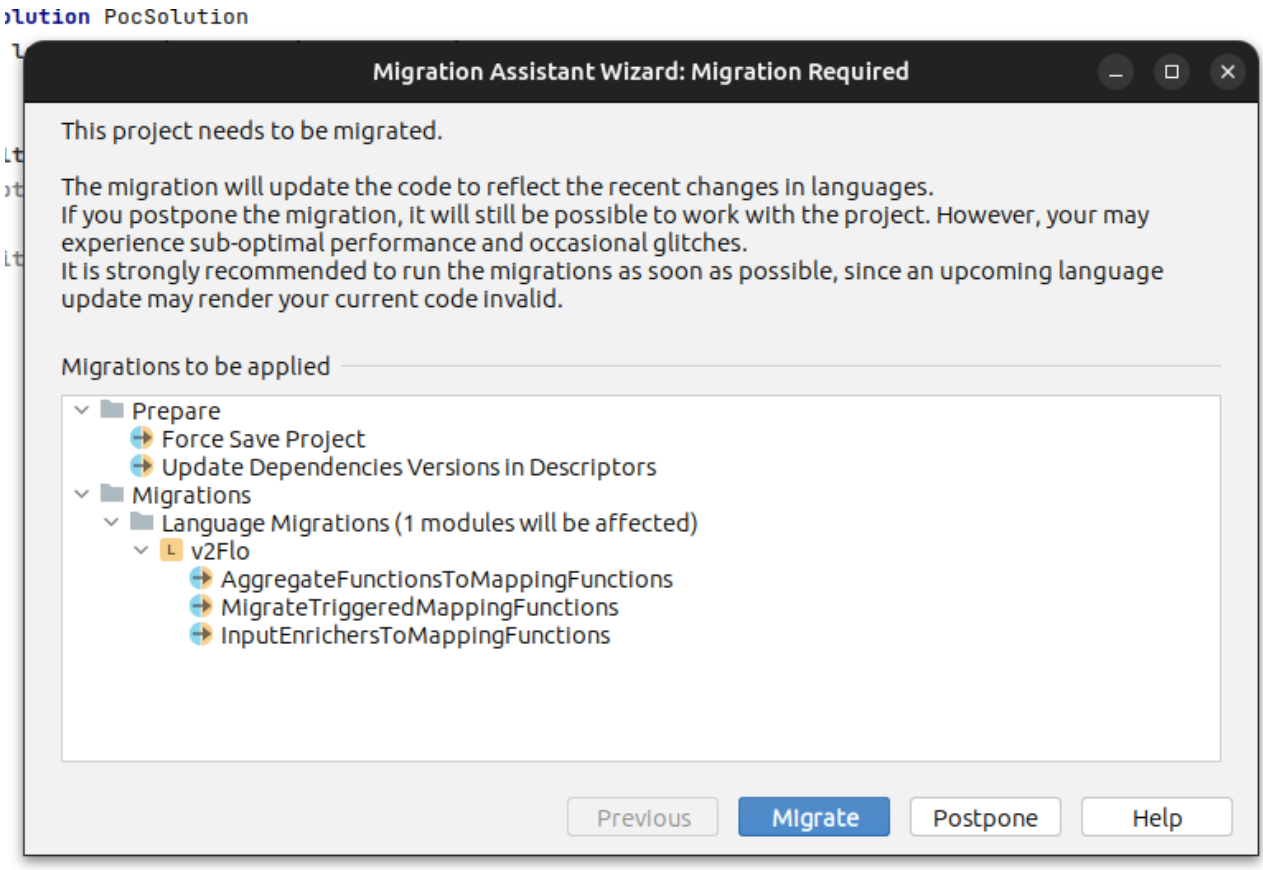
With the changes above completed, you are now ready to perform a standard maven build against your project. This build may fail because the model has not yet been migrated. You may get a build failure such as this:

```
[ERROR] Failed to execute goal com.iconsolutions.plugins:icon-mps-runner:3.0.4:modelcheck (com.iconsolutions.ipf.core.flow_flo-mps-plugin-base-tile_3.17.0__modelcheck) on project mps: Command execution failed.: Process exited with an error: 255 (Exit value: 255) -> [Help 1]
```

Or a compilation error similar to:

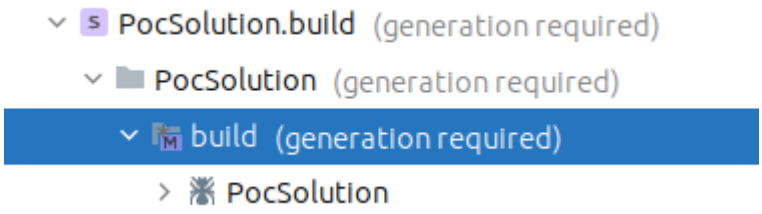
```
[INFO] --- compiler:3.12.1:testCompile (default-testCompile) @ domain ---
[INFO] Recompiling the module because of changed dependency.
[INFO] Compiling 1 source file with javac [debug parameters release 17] to target/test-classes
[INFO] -------------------------------------------------------------
[ERROR] COMPILATION ERROR :
[INFO] -------------------------------------------------------------
[ERROR] /path/to/domain/src/test/java/com/mycorp/ipf/payments/debtor/debtor_ct/behaviour/DebtorCtBehaviourTest.java:[21,53] package com.mycorp.ipf.payments.debtor.execution.testfw does not exist
```

To resolve this, open your project in MPS (remembering to now use `2022.3.1`) and you should be prompted to migrate your solution, click `Migrate`.



> If the Migration Assistant Wizard pop-up doesn't appear, it's likely that you don't have the `Migration Support` plugin enabled in MPS. This will be apparent if you don't see `Migration` in the toolbar. For guidance on how enable this plugin, see theEnable Migration Support plugin in MPS section.

In the navigator panel, you should see that regeneration of the build script is required:



Right click and select `Rebuild Model [Your Model Name]`. Now rebuild using Maven and your project should correctly build.

> If your domain model is large, the domain build may fail with a Stack Overflow exception. For guidance on how to fix this issue, see the Fix Stack Overflow build exception section.

> Build scripts are now only required for projects that are required to be themselves used as dependencies in other MPS projects. If this is not the case for your project, you can simply delete the build solution from your MPS project.

Java17 Updates

Removal of `JUnitReportingRunner`

The Test Framework has been upgraded to use a newer version of JBehave, and this has allowed for the removal of `com.github.valfirst.jbehave.junit.monitoring.JUnitReportingRunner`.

When compiling against the new 2024.1.0 BOM, your Test Framework runners might fail to compile if they are annotated with:

```
@RunWith(JUnitReportingRunner.class)
```

This annotation is no longer required and can be removed.

Changes as a result of Spring 6/Spring Boot 3 upgrade

One of the main goals of the IPF Java 17 upgrade was to be able to migrate to Spring 6 and Spring Boot 3 to benefit from faster vulnerability and CVE fixes. Spring projects have made a number of breaking changes that are unrelated to IPF, but which you may encounter as part of any such Spring upgrade:

- Moving from Java EE (`javax.*`) to Jakarta EE (`jakarta.*`) internally. Here's a table that may be helpful:

Component	Package was	Package now
Jakarta Servlet	`javax.servlet.*`	`jakarta.servlet.*`
Jakarta Validation	`javax.validation.*`	`jakarta.validation.*`
Jakarta Annotations	`javax.annotation.*`	`jakarta.annotation.*`
Jakarta Messaging	`javax.jms.*`	`jakarta.jms.*`

- Spring Web: Changing the default trailing slash matching behaviour to not be enabled by default (link)
- MongoDB Client 4.11.x: this allows using features introduced up to and including MongoDB 7.0 (and is backward compatible to 3.6)
- Spring has removed the `LocalVariableTableParameterNameDiscoverer` in 6.1. This means that Spring will no longer look into the `LocalVariableTable` as a hint. This might result in the Spring `ApplicationContext` now failing to start with errors similar to:

  ```
  Caused by: org.springframework.beans.factory.NoUniqueBeanDefinitionException: No qualifying bean of type
  'com.myorg.MyBean' available: expected single matching bean but found 2: myBean1,myBean2
  ```

  To overcome this, you will need to add this to your root POM to generate method parameter metadata for reflection:

  ```
  <plugin>
     <groupId>org.apache.maven.plugins</groupId>
     <artifactId>maven-compiler-plugin</artifactId>
     <configuration>
  ```

```
        <parameters>true</parameters>
    </configuration>
</plugin>
```

See Parameter Name Retention for more information on this topic.

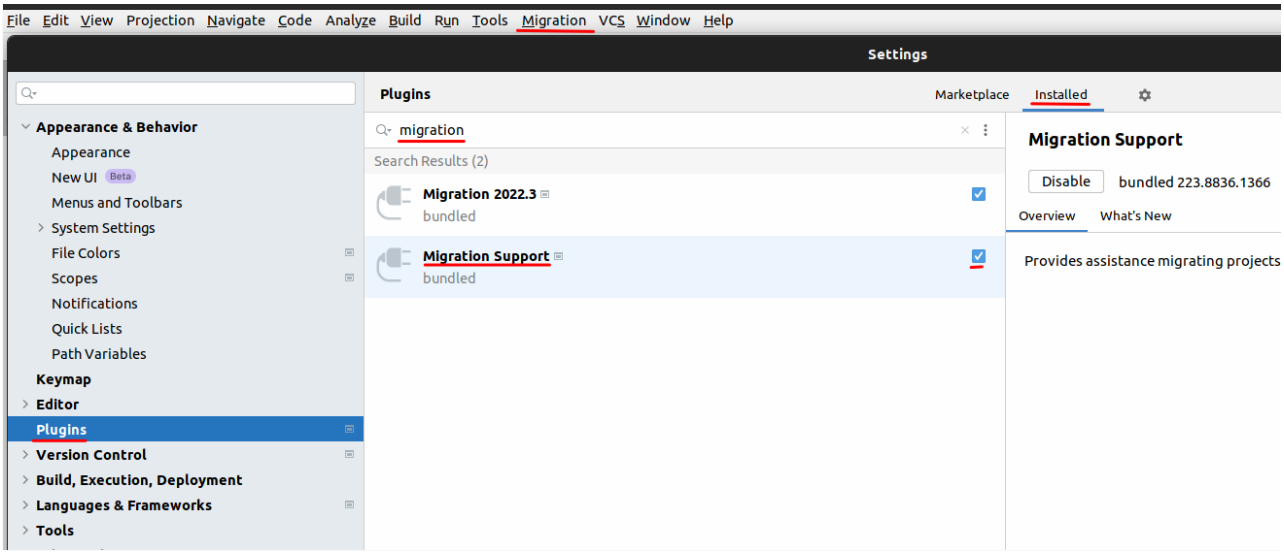
- Spring has also removed support for registering auto-configurations in `spring.factories` using the `org.springframework.boot.autoconfigure.EnableAutoConfiguration` key, in favour of using the `META-INF/spring/org.springframework.boot.autoconfigure.AutoConfiguration.imports` file introduced in Spring Boot 2.7.

For more information on Spring 6+/Spring Boot 3+, consult the Spring Boot 3.0 Release Notes and Spring Boot 3.0 Migration Guide.

Appendix

Enable Migration Support plugin in MPS

To enable the `Migration Support` plugin in MPS, go to File > Settings > Plugins and search for `migration` in your list of Installed plugins. Tick the `Migration Support` plugin to enable it.



Fix Stack Overflow build exception

If your domain module build fails due to a Stack Overflow exception, add the following plugin to the `<build>` section of the `mps` submodule pom.xml:

```
<plugin>
   <groupId>com.iconsolutions.plugins</groupId>
   <artifactId>icon-mps-runner</artifactId>
   <version>3.0.4</version>
   <configuration>
       <vmArgs>
           <vmArg>-Xss256m</vmArg>
       </vmArgs>
   </configuration>
</plugin>
```

Release Notes for IPF-2024.1.2

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-1-2/release-IPF-2024-1-2.html

Release Notes for IPF-2024.1.2

This page details everything required to get going on IPF Release 2024.1.2 made available on 6th June 2024.

> There is not a 2024.1.1 release and this release (2024.1.2) is the first hotfix release for 2024.1.0

Change Spotlight

Fix Spotlight

- Fixed issue where, in some rare cases, generation of flow event handlers fails when building the domain.

Release Details

Binaries

The core binaries are available from IPF-Releases.

Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Akka/Alpakka binaries from 2.9.x onwards are no longer on Maven Central and are available from the Akka Library Repository:

```
<repository>
  <id>akka-repository</id>
  <name>Akka library repository</name>
  <url>https://repo.akka.io/maven</url>
</repository>
```

Archetype

The latest version of the Archetype is **5.0.9**

Developer App

The latest version of the Developer App is **2.1.2**

2024.1.2 Jar/Pom list

The Jars and Poms for 2024.1.2 and the associated versions are listed here: 2024-1-2-artifacts.xlsx

Archetype

The latest version of the Archetype is 4.0.2

Developer App

The latest version of the Developer App is 1.14.6

Migration Steps for IPF-2024.1.2

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-1-2/release-IPF-2024-1-2-migration.html

Migration Steps for IPF-2024.1.2

Version Updates

To migrate to `2024.1.2`, please perform the following steps:

- Update your BOM version to the new release version `2024.1.2`

```
 <parent>
    <groupId>com.iconsolutions.ipf</groupId>
    <artifactId>ipf-bom</artifactId>
    <version>2024.1.2</version>
</parent>
```

- Run a Maven build to retrieve all the latest dependencies.

Release Notes for IPF-2024.1.3

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-1-3/release-IPF-2024-1-3.html

Release Notes for IPF-2024.1.3

This page details everything required to get going on IPF Release 2024.1.3 made available on 21st June 2024.

Change Spotlight

Fix Spotlight

- Fixed issue where, in some cases, MPS compilation can fail with a "code too large" error during generation of BDD test classes - (*PAY-10909*)

Release Details

Binaries

The core binaries are available from IPF-Releases.

Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Akka/Alpakka binaries from 2.9.x onwards are no longer on Maven Central and are available from the Akka Library Repository:

```
<repository>
  <id>akka-repository</id>
  <name>Akka library repository</name>
  <url>https://repo.akka.io/maven</url>
</repository>
```

Archetype

The latest version of the Archetype is **5.0.9**

Developer App

The latest version of the Developer App is **2.1.2**

2024.1.3 Jar/Pom list

The Jars and Poms for 2024.1.3 and the associated versions are listed here: 2024-1-3-artifacts.xlsx

---

Migration Steps for IPF-2024.1.3

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2024-1-3/release-IPF-2024-1-3-migration.html

Migration Steps for IPF-2024.1.3

Version Updates

To migrate to `2024.1.3`, please perform the following steps:

- Update your BOM version to the new release version `2024.1.3`

```
 <parent>
    <groupId>com.iconsolutions.ipf</groupId>
    <artifactId>ipf-bom</artifactId>
    <version>2024.1.3</version>
</parent>
```

- Run a Maven build to retrieve all the latest dependencies.

Release Notes for IPF-2023.4.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-4-0/release-IPF-2023-4-0.html

Release Notes for IPF-2023.4.0

This page details everything required to get going on IPF Release 2023.4.0 made available on 9th February 2024.

Details

Binaries

The core binaries are available from IPF-Releases.

The Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Archetype

The latest version of the Archetype is 4.0.2

Developer App

The latest version of the Developer App is 1.14.6

Features and Changes

Here's what's new and changed in 2023.4.0

Fix Spotlight

- Fixed Journal delete and update speed issues
- Added a timestamp to the entries in the Journal Collection
- Final state ESBs now schedule their own passivation on recovery

Breaking Changes

Java 17

> From release V2024.1 (May 2024), IPF will be using Java 17/Spring 6/Spring Boot 3 for all IPF components. IPF will no longer be compatible with Java 11, and you will need to use Java 17 at design, build and run time.

DSL Changes

The DSL implementation has been updated to allow stronger reuse. There are two major changes in this regard:

Aggregate Functions & Input Enrichers

Aggregate functions and input enrichers were used to allow data to be manipulated during flow processing. In essence, both of these functions perform a mapping from one set of data to another. They have therefore now been replaced by a single 'Mapping Function' concept.

When using a mapping function instead of the previous input enricher, everything is the same. You define the mapping function in the mappings section:

Mapping Function Definitions

Define the functions that will map business data from one set to another for use during the flow processing.

	Name	Description	Input Data	Output Data
1	TestAggregateFunction	test		

Add Function

And then apply which input enrichments to put the mapping on:

Input Behaviour

Define how inputs should processed and what domain events they translate to.

	Input	Response Code	Perform Enrichment	Event Selection
1	bpNotificationResponse	<Any Response Code>	TestAggregateFunction	Beneficiary Notified

For aggregate functions, the slight difference is that they are now defined in the mappings section above but then applied at the event level:

	Name	Description	Business Data	Generate Aggregate Data
1	Test Event	tbc		TestAggregateFunction

In addition to the existing capabilities provided by the old aggregate functions and input enrichers, we also get the following capabilities:

- Mapping functions can be defined within a 'Mapping Library' that is not restricted to a single flow. This means that they can be used across multiple flows or versions of a flow.
- Mapping functions can now be used to define a mapping required to execute an outbound action without impacting the rest of the flow.

RequestReplyConnector Changes

The signature for request reply connectors have changed so that the resulting object is now wrapped in a 'Response' object:

```
public class RequestReplySendConnector<REQ_D, REQ_T, REP_D, REP_T> implements SendingConnector<REQ_D, Response<REP_D>>
```

This response object contains the original response entity in the 'value' field but also provides access to the application's processing context.

ActionNames deprecated

`ActionNames` has been removed and has been replaced with `[name of external domain]ActionNames`.

Any calls to `Action.getActionName().name()` in your solution will need to be replaced with `Action.getActionName()`.

ODS

New Features

Customisable Summaries

- Allows the default built-in summary field mappings to be customised per solution.
- A library is produced for downstream clients/client-teams against which, custom summary mappings can be built and easily tested. Documentation is provided describing how to "plugin" the custom mappings into ODS Ingestion.

IPF Archiver

- The IPF Archiver is an optional add-on, intended to be used alongside ODS Ingestion, that produces archive bundles for each unit of work that meets the criteria for archiving, e.g. it has reached a terminal state and is outside a configured grace period.

FX Support

- Two new Core PDS types have been added: AdditionalIdentifier and Fx.
- Additional identifiers are now exported from a Process Flow Event as a PDS object instead of as a Custom Object.
- Fx PDS Objects are used to handle foreign exchange data. These are mapped to new Summary fields which are returned as part of the ODS Inquiry summaries APIs. These new fields are:
    - convertedTransactionAmount
    - convertedTransactionAmountCurrency
    - exchangeRate
- Both new Core PDS objects can be exported to ODS via a Process Flow Event or via the direct PDS processing-data egress exporter

Large Message Log Support

- Message log entries now support a reference field. This new field is used as an identifier to a Message Log that is stored externally sure to its file size.

- IPF ODS can handle this new field, storing it as part of the MessageLogEntry document, and returning it as part of the message log ODS Inquiry APIs.
- Additionally, a message log direct data exporter has been added to IPF Processing Data. Allowing for message log entries to be exported without implementing the MessageLogger interface

CosmosDB TTL Purging (database change)

- The default purging implementation that we use for MongoDB does not perform well in a CosmosDB environment, and it also has a large impact on ODS Ingestion throughput.
- This CosmosDB-specific implementation is provided as an alternative, and is entirely optional.
- This implementation should cost less overall, and have a smaller impact on ODS Ingestion throughput, but with the added complexity that "housekeeping" jobs must be run (automated and configurable) to maintain feature-parity with the standard purging implementation.

Notable Changes and Improvements

Deprecate custom objects in IPF Processing Data

- As part of migration to PDS objects, the option to categorise business data elements in a flow as CUSTOM has been removed. They must now be categorised as PROCESSING_DATA_STRUCTURE.
- Existing sample flows have been migrated and IPF Processing Data no longer produces custom object data structures. Additionally, this means that IPF Processing Data no longer produces custom objects in duplicate of PDS objects.
- Allow clients to specify MDS and PDS identifiers.
- When exporting a MDS data structure, a id and parentId can now be provided alongside the MDS.When specified, these Ids are mapped to the top level MDS Object that is produced by IPF Processing Data.If not specified, the Ids are generated.
- A PDS object is uniquely identified by its name and unitOfWorkId.When directly exporting a PDS data structure, a name can now be provided to the exporter.If not specified, the name is generated from the PDS data structure.

Introduce the unitOfWorks collection (database change)

- Added a new ODS collection: unitOfWorks. Used to track metadata for a unit of work as opposed to a Summary which is used as a business data view. Updates have been made to utilise this collection instead of the summaries collection for internal processes such as purging, and in future, archiving.
- This collection is not exposed as part of the ODS Inquiry APIs.
- In addition, configuration has been added to allow for ODS Ingestion Summary functionality to be disabled. Meaning ODS Ingestion can be deployed without Inquiry and summaries while still keeping a view of a unit of work’s metadata within the unitOfWorks collection.

GUI

GUI SDK improvements

- Angular 16 upgrade
    - A needed upgrade to maintain 'N-1' with angular’s release schedule.
- Cypress.
    - Added cypress testing to our modules in order to properly test component functionality.
    - Currently only set up for bank-filtering and HTM flows but more will be added moving forward with other tickets.
- Move the common module to the ops-gui-framework repository.
- Jest tests performance issue resolved.
- Resolved long lasting '--legacy-peer-deps' flag issue.
- More reusable components added to help improve the speed of development.

GUI support to allow BIC / Bank / Currency filtering by operators

- Added bank filtering flow
- Some general 'dynamic form' improvements to help ease of development and speed at which we can build new features.
- Refactor of CSM Agent and Agent Currency to align more with our other settings modules.
- Added the ability to search and view FX data points from the GUI.

    - Also added these data points to the results table.

SEPA CT

- Building upon the foundation laid by the previously delivered MVP.
- Integrate non-payment messages with the existing SEPA CT Outbound and Inbound processes.
- Implement comprehensive validation checks for incoming and outgoing non-payment messages to ensure adherence to SEPA standards. Facilitate efficient error reporting and resolution for improved system reliability.
- Support for additional message types, including camt.056 (Bank-to-Customer Cash Management) and pacs.028 (Financial Institution-to-Customer Statement).
- Implemented mechanisms for the effective handling and processing of non-payment messages, ensuring compatibility with existing workflows.
- Error-handling to manage exceptions and maintain the integrity of data transmission.

CSM Services

Added a new field `externalRequestBody` which is available on the following types sent from the following instant payment CSM Services:

- RT1
- FedNow
- T2

This field is available on the following messages from the CSM Service:

- `ReceivePaymentRequest`
- `ReceivePaymentSettledRequest`
- `ReceivePaymentStatusInquiryRequest`
- `ReceiveRecallRequest`
- `ReceivePositiveAnswer`
- `ReceiveNegativeAnswer`

These messages will be available on the various `handleXxx` methods as outlined in Use the CSM Service Client Library .

The respective methods where the `externalRequestBody` will be available are therefore:

- `handleReceivePaymentRequest`
- `handleReceivePaymentSettledRequest`
- `handleReceivePaymentStatusInquiryRequest`
- `handleReceiveRecallRequest`
- `handleReceivePositiveAnswer`
- `handleReceiveNegativeAnswer`

Migration Steps for IPF-2023.4.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-4-0/release-IPF-2023-4-0-migration.html

Migration Steps for IPF-2023.4.0

Version Updates

DSL Migration

> You will need to migrate to the new version of the DSL before being able to use the new release in your application.

To migrate from 2023.3.0, please perform the following steps:

- Update your BOM version to the new release version `2023.4.0`:

```
<parent>
   <groupId>com.iconsolutions.ipf</groupId>
   <artifactId>ipf-release-core-bom</artifactId>
   <version>2023.4.0</version>
</parent>
```

- Update all the flo versions within the domain folders to `2.2.23`. Namely, in "docs", "domain", "external-libraries", "mps", "sampleapp" and "test" modules, update to look like:

```
<parent>
   <groupId>com.iconsolutions.ipf.core.flow</groupId>
   <artifactId>flo-starter-<modulename></artifactId>
   <version>2.2.23</version>
   <relativePath></relativePath>
</parent>
```

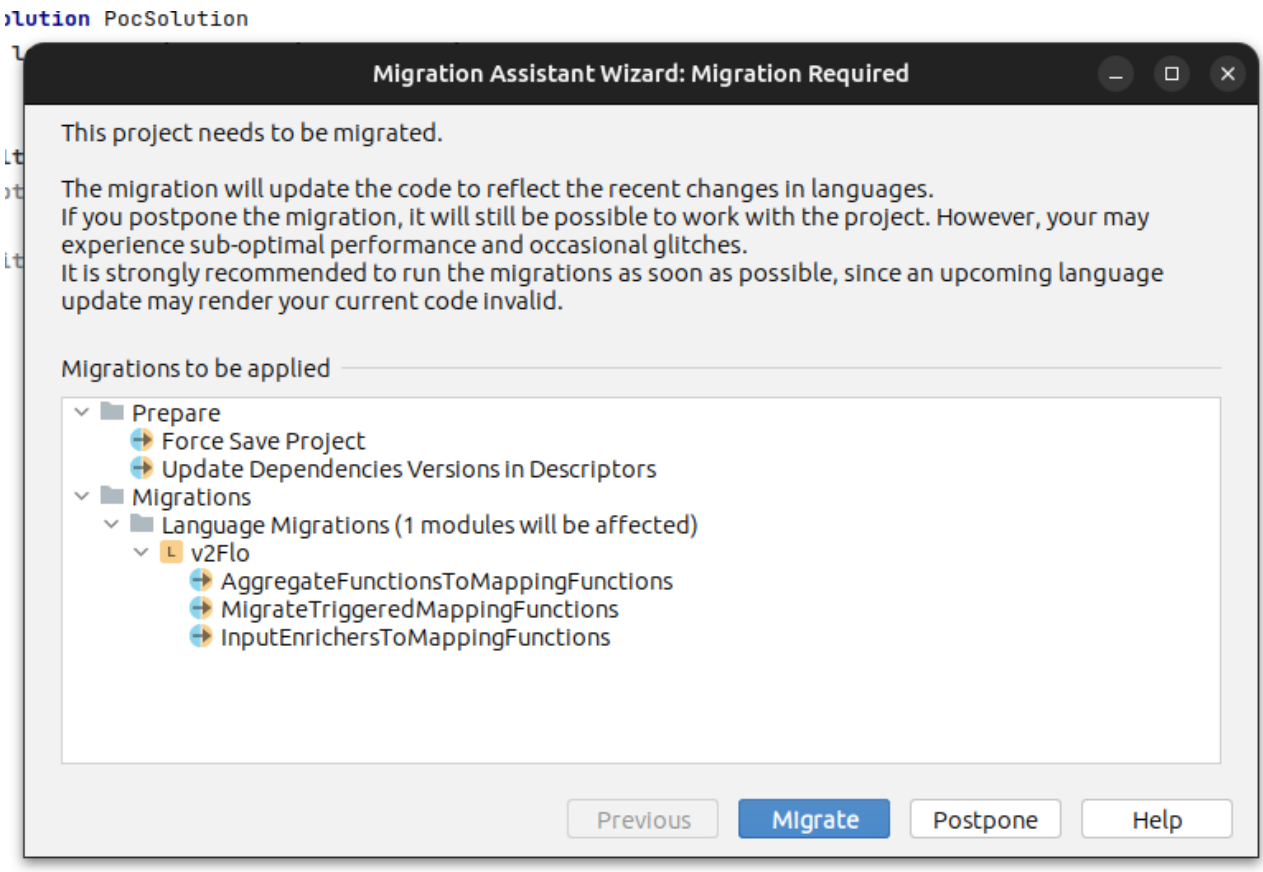
If using any of the icon AOM's (see AOM Details) then you need to add the new aom bom as a dependency:

```
<dependency>
   <groupId>com.iconsolutions.ipf</groupId>
   <artifactId>ipf-release-aom-bom</artifactId>
   <version>2023.4.0</version>
   <type>pom</type>
   <scope>import</scope>
</dependency>
```

- Run a Maven build to retrieve all the latest dependencies. This maven build is expected to fail because the test model domain will not exist. You should get an error similar to:

```
package your.package.structure.testfw does not exist
```

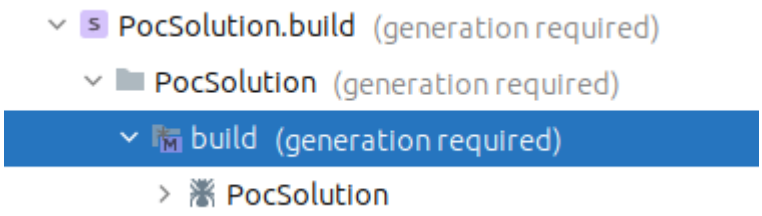
- Open your project in MPS and you should be prompted to migrate your solution, click `Migrate`.



- Next if you open your build solution in the YourApplication.build project you should see the `de.itemis.mps.extensions` entry in red. Simply remove this line.

```
dependencies:
  mps (artifacts location $mps_home)
  v2Flo (artifacts location $plugin_home/v2Flo)
  de.itemis.mps.extensions (artifacts location $plugin_home/de.itemis.mps.extensions)
```

- Now in the navigator panel you should see that regeneration of the build script is required:



- Right click and select `Rebuild Model 'Your Model Name'`
- Now rebuild using maven and your project should correctly build.

Generated Code Migration

Input Initiation Port

The DSL now makes reference to the model in which the generated input initiation port exists.

Therefore, the migration utility run in the DSL section will have converted the `InputInitiationPort` to `YourModelNameInitiationPort`.

Aggregate Functions & Input Enrichers

The enhanced capabilities as described in the overview section for mapping capabilities have led to some changes in the naming of the generated code.

The migration utility run in the DSL section will have converted all aggregate functions and input enrichers to the new style `Mapping Functions`. The following table defines the expected changes:

Class Description	Old Name	New Name
Aggregate Function's Port	YourFlow*AggregateFunction*Port	YourFlow*Mapping*Port
Aggregate Function Parameters	YourAggregateFunctionNameForFlowYourFlow*AggregateFunctionParameters*	YourAggregateFunctionNameForFlowYourFlow*MappingParameters*
Aggregate Function Output	YourAggregateFunctionNameForFlowYourFlow*AggregateFunctionOutput*	YourAggregateFunctionNameForFlowYourFlow*MappingOutput*
Input Enricher's Port	YourFlow*InputEnricher*Port	YourFlow*Mapping*Port
Input Enricher Parameters	YourInputEnricherNameForFlowYourFlow*InputEnricherParameters*	YourInputEnricherNameForFlowYourFlow*MappingParameters*
Input Enricher Output	YourInputEnricherNameForFlowYourFlow*InputEnricherOutput*	YourInputEnricherNameForFlowYourFlow*MappingOutput*

In addition, the migration to the new style `Mapping Functions` will also change the package names in which the above classes now reside. For example:

- `com.your.project.model_name.flow_name.aggregatefunction.YourAggregateFunctionNameForFlowYourFlowAggregateFunctionOutput;`

becomes

- `com.your.project.model_name.mapping.YourAggregateFunctionNameForFlowYourFlowMappingOutput;`

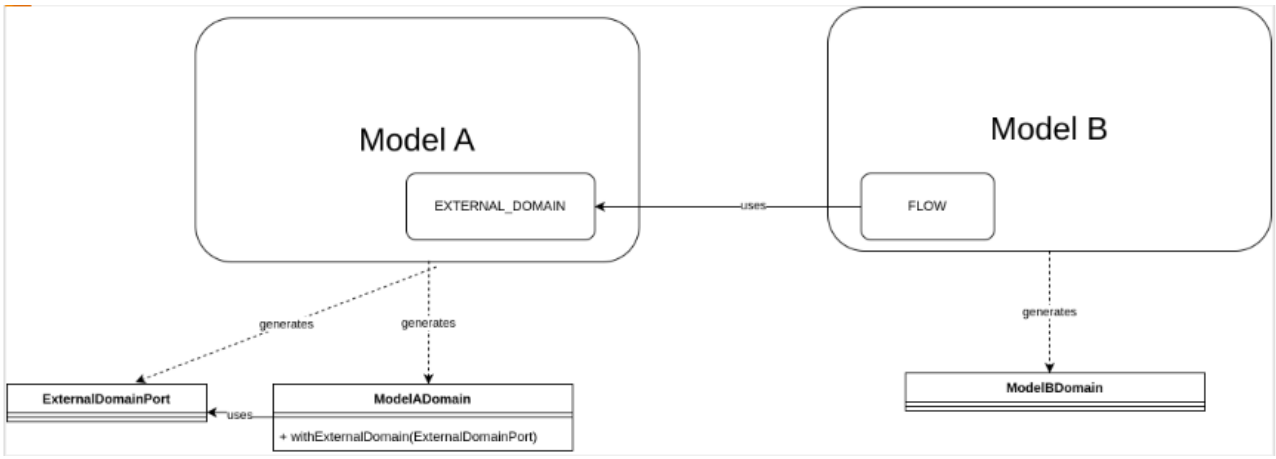
> There is no distinction now between an aggregate function and an input enricher from the generated code viewpoint. Therefore, if your flow had both an aggregate function and an input enricher, previously you would have defined a port for both whereas now a single port will be generated containing all your methods.
>
> The `YourFlowDomain` class will therefore need to be updated. Whereas previous it would have a withXXX method for both aggregate and input enrichment, it will now have a single method for the new mapping port.

Remote Models

An important change is now present when using remote models in MPS, so for example if you define your flows in one model and your external domains in a separate model.

Previously all generated code would be generated from the viewpoint of the flow based model. This meant that the implementation of the external domain logic would have to be performed in the flow implementation.

Now however, code is generated according to the model it has been defined in. Imagine you have two models, Model A containing an external domain called 'EXTERNAL_DOMAIN' and Model B containing a flow that uses the external domain.



Here the key thing to understand is that the elements in ModelA are generated within the scope of model A and similarly for ModelB. This means that there will now be the two separate model domain classes and BOTH are needed to be implemented in order for the flow to work.

The major benefit of this change is that we can now package and reuse remote models together with their implementations.

Release Notes for IPF-2023.4.1

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-4-1/release-IPF-2023-4-1.html

Release Notes for IPF-2023.4.1

This page details everything required to get going on IPF Release 2023.4.1 made available on 25th February 2024.

Details

Binaries

The core binaries are available from IPF-Releases.

The Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Archetype

The latest version of the Archetype is 4.0.2

Developer App

The latest version of the Developer App is 1.14.6

Features and Changes

Here's what's new and changed in 2023.4.1

Fix Spotlight

- Restructured Aggregate Functions in the DSL, which now reference Mapping Functions introduced in 2023.4.0
- Fixed issue where MPS compilation would fail with a "code too large" error during generation of the `YourFlowNameActionRevivalProcessor` class when the number of performed actions in the flow exceeded a threshold value.

Migration Steps for IPF-2023.4.1

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-4-1/release-IPF-2023-4-1-migration.html

Migration Steps for IPF-2023.4.1

Version Updates

> It is recommended that you upgrade directly from `2023.3.0.x` to `2023.4.1` if your solution uses aggregate functions.

To migrate to `2023.4.1`, please perform the following steps:

- Update your BOM version to the new release version `2023.4.1`

```
<parent>
   <groupId>com.iconsolutions.ipf</groupId>
   <artifactId>ipf-release-core-bom</artifactId>
   <version>2023.4.1</version>
</parent>
```

- Update all the flo versions within the domain folders to `2.3.2`. Namely, in "docs", "domain", "external-libraries", "mps", "sampleapp" and "test" modules, update to look like:

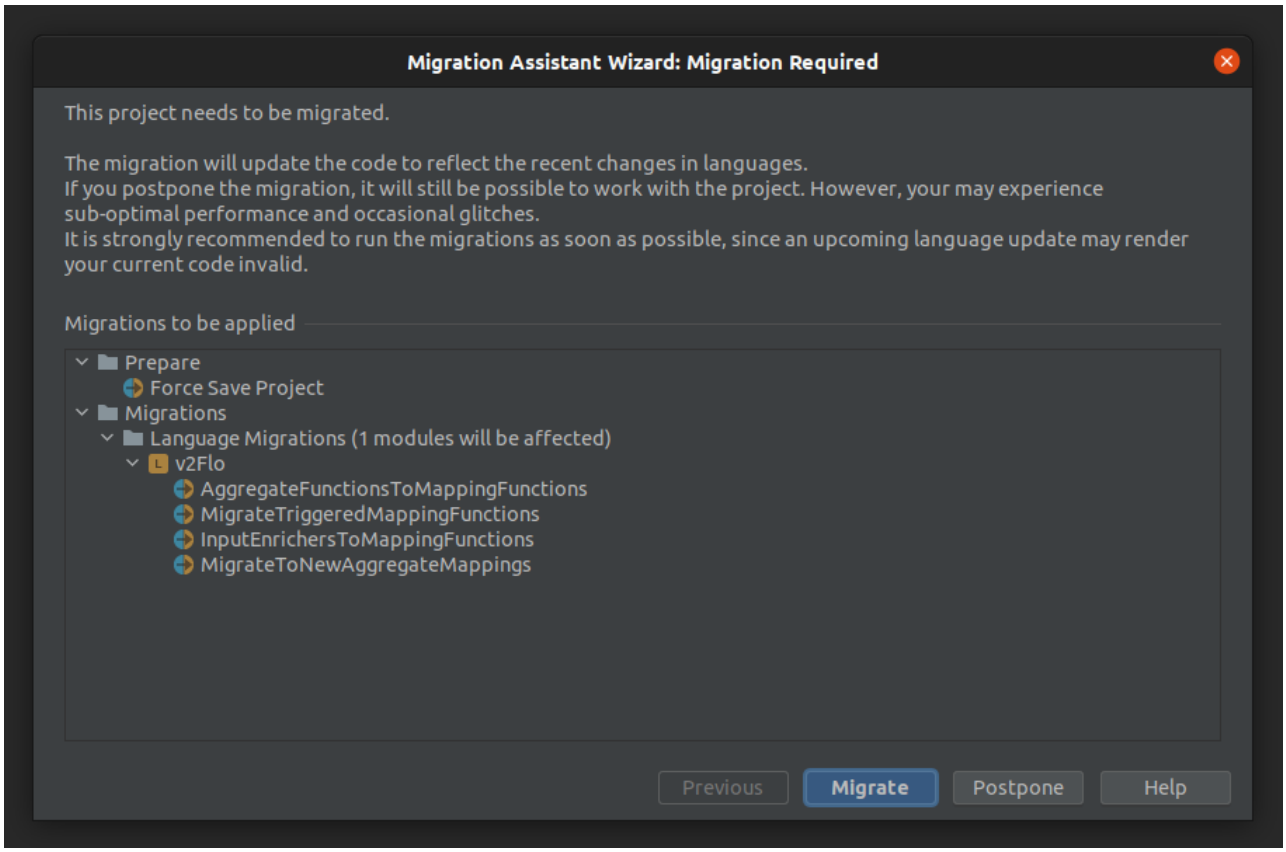
```
<parent>
   <groupId>com.iconsolutions.ipf.core.flow</groupId>
   <artifactId>flo-starter-<modulename></artifactId>
   <version>2.3.2</version>
   <relativePath></relativePath>
</parent>
```

- Run a Maven build to retrieve all the latest dependencies.

DSL Migration

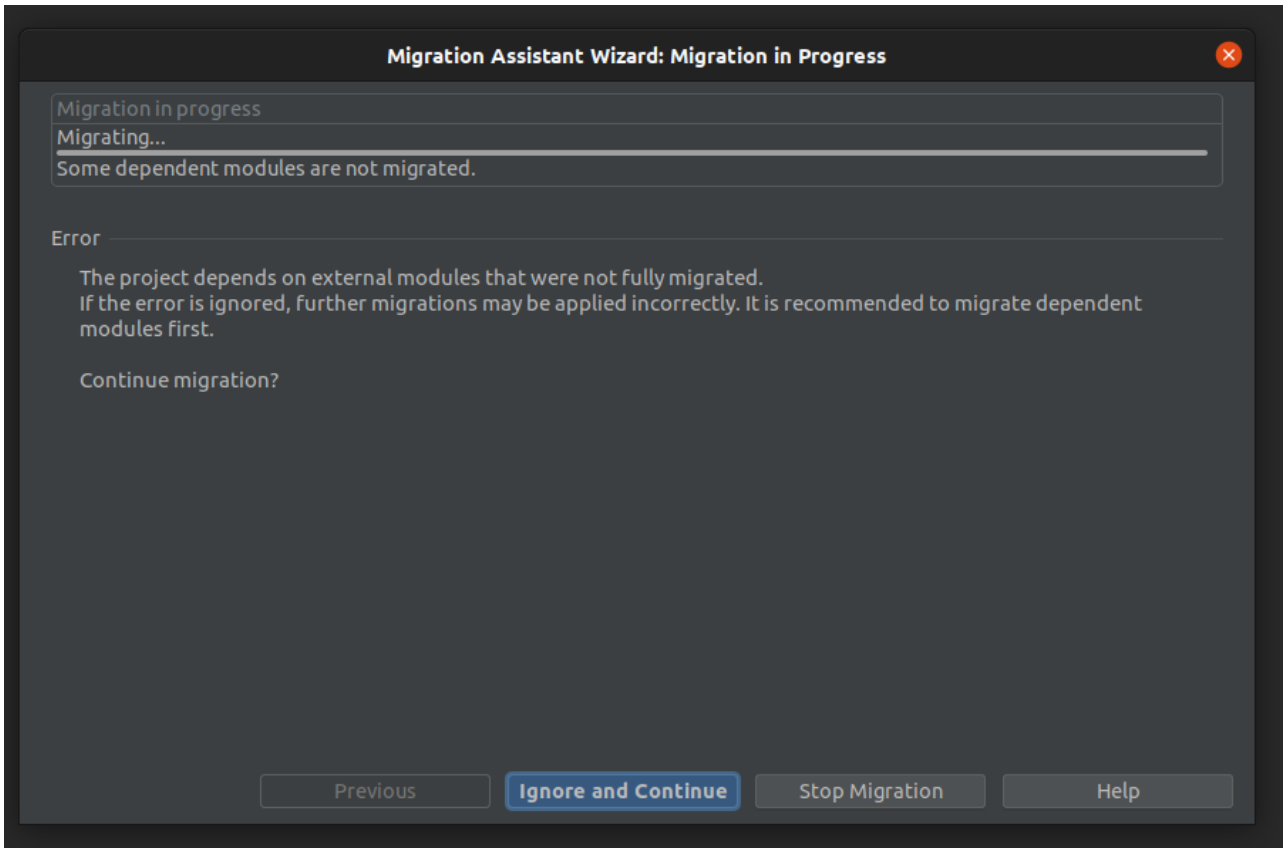
> If upgrading from `2023.3.0.x`, please follow the steps in IPF 2023-4-0 DSL Migration section, but replace the step for migrating your solution in MPS with the ones in this section.

- When you are prompted to migrate your solution in MPS, click `Migrate`. The 'migrations to be applied' may look different to that shown below, as they depend on which version you are migrating from.

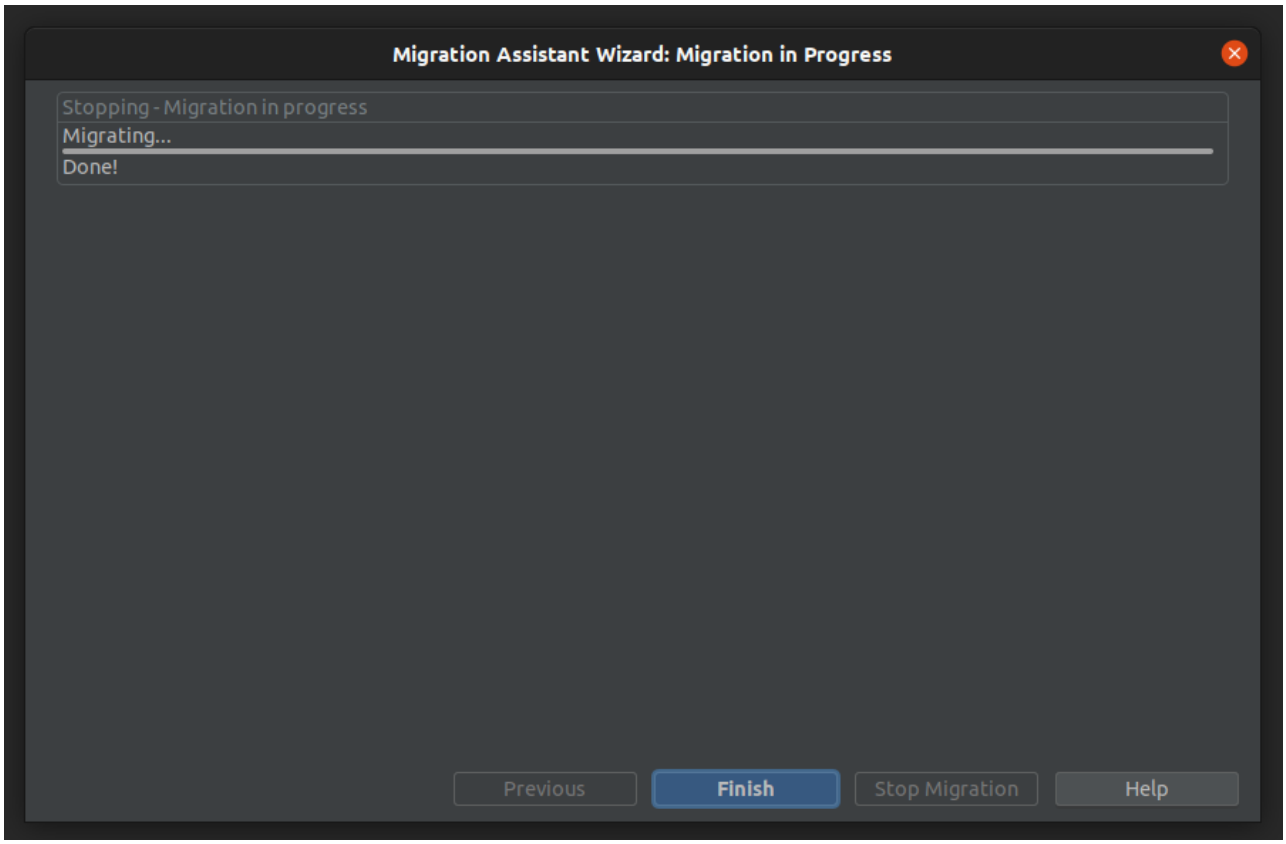


- During the migration process, you may be alerted, more than once, that an error has occurred during the migration process. Each time

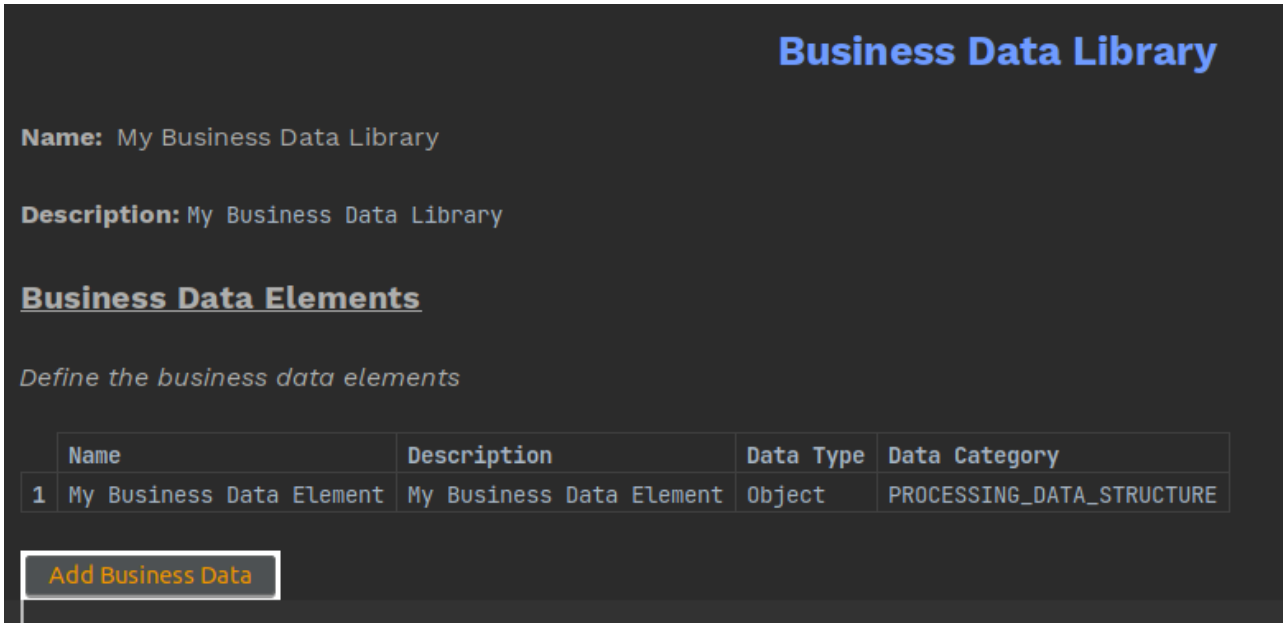
this occurs, click on `Ignore and Continue` to continue the migration process.



- When the migration process has completed, click `Finish`.



As part of the migration process, any `CUSTOM` business data elements will have been migrated to the new `PROCESSING_DATA_STRUCTURE` data category type. More detail about the deprecation of the `CUSTOM` data category type can be found in IPF 2023-4-0 Release Notes: Notable Changes and Improvements - Deprecate custom objects in IPF Processing Data.



Generated Code Migration

> If upgrading from `2023.3.0.x`, the DSL migration steps in the IPF 2023-4-0 Generated Code Migration section will need to be followed in addition to what is detailed in this section.

Domain Events in Aggregate Functions

Domain Events in aggregate functions are now available through the new mapping parameters `MappingContext` object. Below shows an example of how a `DomainEvent` can be accessed from the mapping parameters of an aggregate function via the `MappingContext`:

```
 DomainEvent event = (DomainEvent) yourMappingParameters.getMappingContext().getEvent().orElseThrow(() -> new
IconRuntimeException("The DomainEvent cannot be accessed from the MappingContext."));
```

> The return type for `yourMappingParameters.getMappingContext().getEvent()` is an `Optional<Event>`.

Domain Events accessed through the `MappingContext` no longer contain references to specific business data elements. Therefore, any previous references to business data elements on a Domain Event in your code must now be added as input/output data in the DSL `Mapping Function Definition` and accessed via the aggregate function mapping parameters.

Release Notes for IPF-2023.3.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-3-0/release-IPF-2023-3-0.html

Release Notes for IPF-2023.3.0

This page details everything required to get going on IPF Release 2023.3.0 made available on 31st October 2023.

Details

Binaries

The core binaries are available from IPF-Releases.

The Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Archetype

The latest version of the Archetype is 2.0.0

Developer App

The latest version of the Developer App is 1.0.75

Features and Changes

Here's what's new and changed in 2023.3.0

Fix Spotlight

- Fix to stop caching the Kafka Producers
- Action Revival now caters for chained timeout state transitions
- Breaking change: DateTime fields in the icon ISO20022 model can support ISO8061 compatible date formats as described by the xs:dateTime complex type details
    - The ISO20022 model now generates to use the IsoDateTime instead of OffsetDateTime
    - Any projects which use those generated models will be broken if they try to access timestamps expecting OffsetDateTime
- AOMs are now available as runnable jars, as well as Docker images
- Null field values are no longer persisted into the journal or snapshot collections

Breaking Changes

Java 17

> Java 17 Migration to take place in May 2024.
>
> From release V2024.1 (May 2024), IPF will be using Java 17/Spring 6/Spring Boot 3 for all IPF components. IPF will no longer be compatible with Java 11, and you will need to use Java 17 at design, build and run time.

New IPF Data Model

It is strongly recommended to migrate to the new model (introduced in the previous release), we will be deprecating and removing the old types in subsequent releases. Details can be found in the docs area in IPF Portal

Human Task Manager

- HTM should give timely feedback to consumer when status has changed
- HTM 500's if processing context is malformed or contains null
- HTM register task due date should not be mandatory
- HTM register task due date should not be mandatory
- Task Details does not contain Execution Operator Notes

Details can be found in the IPF Developer Docs - Human Task Manager

Bulk HTM

Enable HTM for Bulk Task execution

- Bulk Task Read side Processor
- HTM - Configurable Event Processors number
- HTM Bulk configurable limit
- API - Bulk Execute (Single End Point)
- Bulk Task Processor - back end asynchronous service
- YAML changes - For Bulk Execute Request and Bulk Allocate Request services from GUI
- YAML changes - For Bulk Execute Request service from GUI (Single End Point)
- Create New Bulk Task Aggregate
- API - Bulk Allocate - Receive request
- API - Bulk Execute - Receive request
- Create synergizing API

Bank Filtering

A new implementation that allows configuration of filtering rules using Dynamic processing settings and the client flows to enquire the rules for the payments being processed to get highest priority rule that matches through exposed API

- Cache Query interface
- Configurable Process Settings structure for BIC and currency filters
- Bank exclusion - Populate the cache data model
- Filter Identification based on BIC, Country Code or Currency Code
- Filter Date Restrictions and Multiple filter hits
- Implement bank exclusion API
- Bank exclusion logic

GUI Improvements

- Adoption of new IPF Business Data Model
- Created configurable summary page for ods-search module
- Improved the reliability of user roles and user processing entity management
- Fixed external messages not being associated to flows correctly
- A user can now see supporting data alongside the payload
- Fixed flow graphs taking up too much space on smaller screens
- Documentation improvements
- Only show processing entity for users selected processing entity
- Update payload box to contain relevant metadata
- Update Global status and event status to be red on failure
- Finished HTM for IPF core and developed initial UBS HTM features
- Details page execution history

ODS

New Data Model Continuation

- Implemented message rules, allowing validation of ISO MDS types.
- Implemented standard product technical PDS types representing data points determined during IPF processing, e.g. Csm, JourneyType, etc.

Custom Summary Mappings

- Summaries are projections/views of a unit of work and are the result of extracting fields from data produced by IPF processing. These mappings from the source IPF processing data to the summary were predefined by ODS and could not be changed.
- This epic allows clients to define their own summary mappings, both from ISO MDS types, and client-specific PDS types, and provides a simple code-based method of implementing and testing custom summary mappings.

Checkpoints

- Enables a full end-to-end view of a single unit of work, where all the data produced for the unit of work includes a "checkpoint" to the thing before it, providing a causal relationship for all unit of work data.
- From the checkpoints we can determine that an incoming message triggered an event, which triggered an action, which triggered an outgoing message.

Summary indicates terminal/failure status

- IPF now produces process flow definitions that indicate the terminal/failure status of global statuses, and ODS now stores these.
- When ODS receives an event for a unit of work, it resolves the terminal/failure status from the relevant process flow definition and updates the summary when the new global state is failure = true and/or terminal = true.
- These additional summary fields are presented in the ODS Inquiry API.

SEPA CT CSM Pack

- The SEPA CSM has been enhanced to handle pacs.008 messages from clients that contain multiple Credit Transfer Transactions as well as responding with "Fully Accepted" pacs.002 messages as a result of processing the end-of-day RSF file.
    - The Validation API has been updated to include Interbank Settlement Date checks
- The Debulker can now be configured to perform xml validation against an XSD prior to full processing and splitting. Files can now also be delivered from an S3 source address
- SEPA CSM Pack handles inbound pacs.008 messages (received from the CSM).

Migration Steps for IPF-2023.3.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-3-0/release-IPF-2023-3-0-migration.html

Migration Steps for IPF-2023.3.0

Version Updates

New IPF Data Model Migration

> You will need to migrate to the new data model before updating to the new BOM (if not done migrating to 2023.2.0). We will be deprecating and removing the old types in subsequent releases. Details can be found in the IPF Developer Docs - Migrating from legacy data model

To migrate from 2023.2.0, please perform the following steps:

- Update your BOM version to the new release version `2023.3.0`:

```
 <parent>
    <groupId>com.iconsolutions.ipf</groupId>
    <artifactId>ipf-release-core-bom</artifactId>
    <version>2023.3.0</version>
</parent>
```

- Update all the flo versions within the domain folders to `1.36.134`. Namely, in "docs", "domain", "external-libraries", "mps", "sampleapp" and "test" modules, update to look like:

```
 <parent>
    <groupId>com.iconsolutions.ipf.core.flow</groupId>
    <artifactId>flo-starter-<modulename></artifactId>
    <version>1.36.134</version>
    <relativePath></relativePath>
</parent>
```

- Run a Maven build to retrieve all the latest dependencies.

Release Notes for IPF-2023.3.0.1

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-3-0-1/release-IPF-2023-3-0-1.html

Release Notes for IPF-2023.3.0.1

This page details everything required to get going on IPF Release 2023.3.0.1 made available on 2 February 2024.

Details

Binaries

The core binaries are available from IPF-Releases.

The Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Archetype

The latest version of the Archetype is 2.0.0

Developer App

The latest version of the Developer App is 1.0.75

Features and Changes

Here's what's new and changed in 2023.3.0.1

Fix Spotlight

- Fixed issue (when using persistent scheduler for retry/timeout scheduling) where IPF would try to cancel already-cancelled timeouts and retries, causing unnecessary database load

---

Migration Steps for IPF-2023.3.0.1

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-3-0-1/release-IPF-2023-3-0-1-migration.html

Migration Steps for IPF-2023.3.0.1

Version Updates

To migrate to 2023.3.0.1, please perform the following steps:

- Update your BOM version to the new release version 2023.3.0.1

```
 <parent>
    <groupId>com.iconsolutions.ipf</groupId>
    <artifactId>ipf-release-core-bom</artifactId>
    <version>2023.3.0.1</version>
</parent>
```

- Run a Maven build to retrieve all the latest dependencies.

Release Notes for IPF-2023.2.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-2-0/release-IPF-2023-2-0.html

Release Notes for IPF-2023.2.0

This page details everything required to get going on IPF Release 2023.2.0 made available on 7 July 2023.

Details

Binaries

The core binaries are available from IPF-Releases.

The Lightbend Telemetry (formerly known as "Cinnamon") binaries are available from IPF Lightbend.

Archetype

The latest version of the Archetype is 1.7.0

Developer App

The latest version of the Developer App is 1.0.50

Features and Changes

Here's what's new and changed in 2023.2.0

New IPF Data Model

- As part of this release we have introduce a new Java representation of the ISO20022 Message model.
- This new model has has various enhancements and will serve the basis for many more capabilities in future release.
- The new data model is structurally equivalent for the existing message representations, so one representation can be converted to the other transparently through Json (A utility classes is provided for this)
- For full details on how to migrate from the legacy model to the new model please see the Migration Guide provide as part of the developer documentation - this includes an OpenRewrite recipe that can be applied to your project to automate most of the type changes.
- One addition additional change is that the accessor methods of properties with the new model now no-longer lazily initiations any empty collections. These method will now return null if the underlying collection IS null.

> You will need to migrate to the new data model before updating to the new BOM. We will be deprecating and removing the old types in subsequent releases. Details can be found in the IPF Developer Docs - Migrating from legacy data model

Human Task Manager

This is a new IPF Product feature designed to manage Payment transactions which fail validations or process of straight through processing (STP) and need to be placed in a manual operations queue for review or action.

The key features of the HTM application are:

- The ability to register tasks to a manual operational team, supplying details which allow them to manage, filter and sort tasks.
- The ability to allocate a task to a specific operator for investigation and actioning.
- The ability to mark a task as having being executed and to enter information relevant to the action taken, which will be logged and available for audit reporting.
- The (optional) ability to require one or more approvals on an executed task.
- Audit history of the steps taken in the workflow for any given task is logged and can be queried.

Details can be found in the IPF Developer Docs - Human Task Manager

Reachability service extensions

Enhance existing IBAN validation service to include validation of bank code against EXCLUSIONLIST published by SWIFT

- An enhancement to existing service GET/iban/deconstruct to include validation against SWIFT published EXCLUSIONLIST as recommended by SWIFT published documentation.

Support import of STEP2 SCT directory published by EBA

- A new feature to support the load of these STEP2 SEPA Credit Transfer directories: SCT Direct Participant Routing Table, SCT Reachable BIC Routing Table.
- This feature loads those EBA published files into the IPF Participant configuration database which is used in determining CSM Reachability (In this case to determine if a Creditor IBAN is reachable via STEP2 SCT).

Enhancement of IPF Settings configuration database and functionality to include fields required for STEP2 SCT

- New fields were added to these existing IPF dynamically configurable settings database to cater for STEP2 SCT functionality: Participant, CSM Agent Selection.
- Existing IPF CSM Reachability functionality was extended to cater for STEP2 SCT functionality.

Support validation of BIC against SWIFT published bicdir2018 file

- A new feature to validate a BIC against SWIFT published bicdir2018 file.
- This feature can be used by clients who require this for their creditor BIC validation on outbound payments.

Enhance existing IBANPLUS file loader to deal with DELTA files published by SWIFT

- SWIFTRef publish IBANPLUS to their subscribers in either FULL or DELTA file formats. IPF IBANPLUS loader file is enhanced to deal with either published format.

GUI Improvements

- Updated GUI to Angular 15
- Config was added so that the ordering of date and amount types when searching in ODS search could be changed.
- Various tech debt fixes including only calling the config endpoint once (formerly it was called many times, when moving about the GUI), the back button in the ODS search module now goes back to the last search performed rather than the previous page.
- MVP Human Task Manger for the GUI
    - Ability to search for tasks with further improvements in development.

ODS

Purging

- ODS deletes old unit of work data when that unit of work hasn't changed within some configured retention period (defaults to 2 years). Deletions are done in small batches (defaults to 40), frequently throughout the day (defaults to every 1 second).
- A set of known final states can be configured (optional) resulting in units of work being deleted when they are in one of those final states. If a unit of work is outside the retention period, but not in one of the final states, it is not deleted.

The impact of frequent deletion has on ingestion performance is still being assessed, and the outcome of this may result in changes to the default configuration for batch deletion size, and frequency.

New Data Model

- ODS supports the new message model, and existing ISO20022 data in ODS is converted to the new message model as required.
- The ODS Inquiry API OpenAPI specification uses the new message model types, with a massive reduction in the number of types defined, and the overall size of the spec.
- ODS can store and version PDS objects, with any structure. The objects might be known IPF types, or unknown client defined types, with whatever structure they require.
- Some standard IPF PDS types have been defined, but they do not currently live in the correct place, and their final structure may change.

Bulker

- Bulking functionality enhanced to allow bulk of bulks creation, where by a parent bulk can itself be made up of (or include) other child bulks
- Auto generated bulks introduced, providing commands to simply call to add a component to a bulk and the bulker can be configure to create that bulk if it doesn't already exist.
- Enhancements to recurring bulk configuration

Migration Steps for IPF-2023.2.0

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-2-0/release-IPF-2023-2-0-migration.html

Migration Steps for IPF-2023.2.0

Version Updates

New IPF Data Model Migration

> You will need to migrate to the new data model before updating to the new BOM. We will be deprecating and removing the old types in subsequent releases. Details can be found in the IPF Developer Docs - Migrating from legacy data model

To migrate from 2023.1.0, please perform the following steps:

- Update your BOM version to the new release version `2023.2.0`:

```
 <parent>
    <groupId>com.iconsolutions.ipf</groupId>
    <artifactId>ipf-release-core-bom</artifactId>
    <version>2023.2.0</version>
</parent>
```

- Update all the flo versions within the domain folders to `1.36.101`. Namely, in "docs", "domain", "external-libraries", "mps", "sampleapp" and "test" modules, update to look like:

```
 <parent>
    <groupId>com.iconsolutions.ipf.core.flow</groupId>
    <artifactId>flo-starter-<modulename></artifactId>
    <version>1.36.101</version>
    <relativePath></relativePath>
</parent>
```

- Run a Maven build to retrieve all the latest dependencies.

Code Updates

App Bootstrapping

When starting your IPF application with no seed nodes or Cluster Bootstrap configured, you may get this exception:

```
java.lang.IllegalArgumentException: No default service discovery implementation configured in `akka.discovery.method`. Make sure
to configure this setting to your preferred implementation such as 'akka-dns' in your application.conf (from the akka-discovery
module).
```

To remedy this, either specify a service discovery method like Kubernetes, or - if this application does not require clustering - specify this block:

```
akka {
  cluster.seed-nodes = ["akka://"${actor-system-name}"@0.0.0.0:"${akka.remote.artery.canonical.port}]
  remote.artery {
    canonical.port = 55001
    canonical.hostname = 0.0.0.0
    bind.hostname = 0.0.0.0
    bind.port = 55001
  }
}
```

The block above will configure a seed node with a specific address, which will disable Cluster Bootstrap.

> Do not use the block above for distributed Akka clusters. This will create a split-brain. Use Service Discovery with Kubernetes instead.

Serialisation

Any use of the *SerializationHelper.CustomModule* has now been moved and renamed. This will automatically be updated when using the SerializationHelper class but will not be updated if using this in other places like the akka serialization configuration.

If you have something like this in your configuration:

```
akka.serialization.jackson {
  jackson-modules += "com.iconsolutions.ipf.core.shared.api.serializer.SerializationHelper$CustomModule"
}
```

it now needs to be replaced with:

```
akka.serialization.jackson {
  jackson-modules += "com.iconsolutions.ipf.core.shared.api.serializer.module.UntypedObjectDeserializationModule"
}
```

> If using the *ipf-common-starter-core* module, which comes as a transitive dependency with the*ipf-write-starter-mongo* module, this has now been added by default, so you do not need to manually configure it as above.

Release Notes for IPF-2023.2.0.1

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-2-0-1/release-IPF-2023-2-0-1.html

Release Notes for IPF-2023.2.0.1

This page details everything required to get going on IPF Fix Release 2023.2.0.1 made available on 31st July 2023.

Details

Binaries

The core binaries are available from IPF-Releases.

The lightbend cinnamon binaries are available from IPF Lightbend.

Archetype

The latest version of the archetype is 1.7.0

Developer App

The latest version of the developer app is 1.0.50.1

Features and Changes

Developer App

- ODS purging functionality added (see documentation)

Migration Steps for IPF-2023.2.0.1

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-2-0-1/release-IPF-2023-2-0-1-migration.html

Migration Steps for IPF-2023.2.0.1

Version Updates

To migrate to 2023.2.0.1, please perform the following steps:

- Update your BOM version to the new release version 2023.2.0.1

```
 <parent>
    <groupId>com.iconsolutions.ipf</groupId>
    <artifactId>ipf-release-core-bom</artifactId>
    <version>2023.2.0.1</version>
</parent>
```

- Run a maven build to pull down all the latest dependencies.

Release Notes for IPF-2023.2.0.2

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-2-0-2/release-IPF-2023-2-0-2.html

Release Notes for IPF-2023.2.0.2

This page details everything required to get going on IPF Fix Release 2023.2.0.2 made available on 6th Sept 2023.

Details

Binaries

The core binaries are available from IPF-Releases.

The lightbend cinnamon binaries are available from IPF Lightbend.

Archetype

The latest version of the archetype is 1.7.0

Developer App

The latest version of the developer app is 1.0.50.2 (note - the developer app version has changed in this fix release)

Features and Changes

Connector

- Fix: Change to KafkaAckReceiveConnectorTransport broker outage recovery mechanism - due to an Alpakka bug preventing proper recovery, replaced per-partition Kafka source with regular one.
- Deprecated: KafkaAckReceiveConnectorTransport partitions setting (`max-partitions` config option) since it is no longer used by the latest implementation of the transport.

Akka Discovery MongoDB

- Fix: Make TTL (time to live) Index for discovery entries configurable
- Fix: Stop incorrect logback.xml from being brought in. This was causing the wrong logging settings to be picked up.

Migration Steps for IPF-2023.2.0.2

Source: https://docs.ipfdev.co.uk/reference/current/release/IPF-2023-2-0-2/release-IPF-2023-2-0-2-migration.html

Migration Steps for IPF-2023.2.0.2

Version Updates

To migrate to 2023.2.0.2, please perform the following steps:

- Update your BOM version to the new release version 2023.2.0.2

```
 <parent>
    <groupId>com.iconsolutions.ipf</groupId>
    <artifactId>ipf-release-core-bom</artifactId>
    <version>2023.2.0.2</version>
</parent>
```

- Run a maven build to pull down all the latest dependencies.

\n

Reason Codes

1 pages in this subsection

Reason Codes

Source: https://docs.ipfdev.co.uk/reference/current/ipf-reason-codes/ipf-reason-codes.html

Reason Codes

This page aims to serve as a dictionary of IPF proprietary reason codes that are returned by different IPF components to convey an outcome of IPF business function.

Reason codes are a standard way to return information on outcome of an IPF business function API call, this could be a remote HTTP API call or Java interface. They may often be used to provide information in addition to an error code, but they may also be used in a successful API response

It is important to note that not all IPF business functions use reason codes and as more business functions start using them, they will be added here so the list of reason codes and usage within IPF can be found in one place.

Reason Code Format

IPF reason codes follow below format

```
IPFXXXnnn
```

Characters	Format	Details
1-3	"IPF"	Constant "IPF" representing these are IPF reason codes.
4-6	3 Uppercase alpha characters representing the function	Representing the business function. These are not a representation of the application/service, but the logical business function. When an application/service supports multiple functions, it would have multiple different business function codes.
7-9	3 digit numeric	Including leading "0"s. Starting at 001 and incrementing as new codes are added.

IPF reason codes are also associated with descriptions that describe the nature of outcome of the business function call.

Reason Codes List

- Business Function - Reachability
- Business Function - Debulker

\n

Glossary

1 pages in this subsection

Glossary

Source: https://docs.ipfdev.co.uk/reference/current/glossary/glossary.html

Glossary

Term	Meaning
ACCP	An ISO20022 payment status - Accepted Customer Profile, this means the customers profile allows the payment to happen.
ACH	An Automated Clearing House (ACH) is a computer-based electronic network used for processing financial transactions between participating financial institutions. They are often referred to as CSMs and while they are the most common type of CSM, it is not true to say all CSMs are ACHs. There is also an ACH in the USA called ACH.
ACSP	An ISO20022 payment status - Accepted Settlement In Process.
ACTC	An ISO20022 payment status - Accepted Technical Checks.
Action	Something that is done as result of a state transition in a flow defined in flow designer.
Action Timeout Events	An Action Timeout event that is configured to occur when a given time period has passed without an action completing. The flow can react to this event to handle the timeout scenario.
Actor	An Akka programming entity that has a state, a behaviour, an address to receive messages and a mailbox to store received messages in the order they arrive. An Actor is responsible for performing a discreet piece of system behaviour, and is the basic unit of the Akka Actor System.
Actor System	An actor system in Akka is like a highly organized company where every employee (actor) specializes in a specific task and communicates strictly via messages.
Adapter	A piece of java code that implements a DSL concept that can't be entirely specified in the DSL (e.g. an external domain adapter will do all necessary format conversion between the flow and the external system).
Additional Event	A type of event that is raised by the flow itself. When an additional event is raised, the system will process it as though it has been received into the application via an instruction or response.
Additional Optional Module (AOM)	On top of the core IPF license, Additional Optional Modules can be utilised; for example ODS or scheme packs.
Additional Optional Services (AOS)	A concept in the SEPA payment schemes. They are optional services supported by some CSMs which members may choose to adopt or not. IPF scheme packs usually cover the mandatory functionality by default and AOS via specific agreement.
ADMI	ADMInistration messages in ISO20022.
Agent	In IPF terminology, an Agent is an entity which the Processing Entity holds a settlement relationship in one or more Transfer Currencies. The Agents and Processing entities agree to settle transactions between themselves through settlement accounts and without the need of another party getting involved for the settlement.
Aggregate	The aggregate is a group of data objects that can be treated as a single unit for the purpose of data changes and Events in the context of a flow. The aggregate is the live, in memory, data store for the payment instance being processed
Aggregate Function	An Aggregate Function performs some kind of logic when an event is received and/or upon the data received in an event for later use in a flow.
Akka	An open source framework for building concurrent, distributed, and resilient message-driven applications on the JVM, via microservice-style components.
Akka Cluster	A group of interconnected actor systems (nodes) that work together as a single, fault-tolerant application.
Akka streams	A toolkit for processing real-time data flows with high levels of reliance.
Alpakka	A toolkit for processing real-time data flows in Akka-based systems.

Amazon Web Services (AWS)	Cloud computing service that can be used as the deployment environment for IPF solutions.
API	Application Programming Interface: a set of clearly defined methods of communication between software components.
Asciidoc	Asciidoc is a lightweight markup language (like Markdown) for writing structured documents. HTML in comparison is also a markup language but is heavyweight. Asciidoc is used in IPF to produce documentation, such as this page you are reading now.
Association ID	The Association ID within the Processing Context associates a point-in-time event (such as a message-log or system event emission etc.) with a local context such as an IPF Processing Flow. This way we can state "this message log was caused by this Flow" rather than just linking to a single unitOfWork that may span multiple services and flows.
Asynchronous	A processing model where tasks operate independently without waiting, as opposed to synchronous.
Automatic Retries	One possible Resiliency Strategy in IPF, seehere for more.
Azure	Cloud computing service from Microsoft that can be used as the deployment environment for IPF solutions.
Backpressure	One possible Resiliency Strategy in IPF. It's a safety mechanism in data systems where a slow consumer signals to fast producers to pause or slow down, preventing overload. See here for more.
Bank Directory Plus	A SWIFT directory product that consolidates comprehensive reference data required by banks and corporates to prepare, validate, and process payments globally. It includes essential details such as BIC (Bank Identifier Code) information, branch data, and clearing system participation. It is being replaced by the new Identifiers Directory.
Bank filtering	A service that offers the means to filter transactions based on criteria defined through dynamically configurable rules. Enables "emergency breaks" on processing of payments that match a particular criteria. See here for more.
BBAN	Basic Bank Account Number. Each country can have its own BBAN format, e.g. in the UK it's sort code & account number, while in the United States of America it's routing number and account number.
Bean	A Java Bean is a reusable software component in Java programming that follows specific conventions. It's essentially a standardized way of creating objects that can be easily understood and used by different parts of a software application or by different applications altogether. Java Beans are designed to be simple, flexible, and easy to work with, making them useful for both developers and software tools.
Behaviour Driven Development (BDD)	A software development approach that promotes the use of a shared, simplifiedlanguage for the specification and testing of systems that deliver verifiable business value. It specifies requirements in a testable format using given, when & then statements.
Beneficiary Bank	The bank that receives funds as part of a payment. Referred to as Creditor Bank in ISO20022.
BOM	Bill of Materials, a concept in Maven that is used to manage and centralise versions of technical dependencies.
BSON	A binary-encoded version of JSON designed for efficiency in data storage and processing.
BU ID	A Business Unit ID is an identifier used to manage access control and permissions within an organization's security framework.
Bulking/De-Bulking	Bulking is a capability which groups one or more payment instructions or messages into a bulk fileBulking is required when the bank sends multiple transactions into the same scheme for processing a specific type of payment (Bulk files) or when it's a scheme mandatory requirement to send a bulk payment message e.g. SEPA Credit transfer. De-bulking is a capability which splits a bulk message into individual instructions.
Business data	A piece of business data that a flow makes use of (e.g. payment initiation, selected CSM, FX deal details etc.).
Business Identifier Code (BIC)	A BIC is universal identifier for financial and non-financial institutions, used for addressing messages, routing business transactions and identifying business parties. The BIC format is defined by ISO standard 9362:2014. BICs are issued by SWIFT in its role as the ISO registration authority. Note: Under earlier versions of the ISO 9362 standard, BIC stood for Bank Identifier Code and this term is still widely used.
CAMT	Cash Account ManagemenT messages from ISO20022, e.g. camt.056 is a request to cancel.

Checkpoints	Refers to an identifier for the last "processing event" that occurred for a given Processing Context. The Checkpoint implementation in IPF captures internal events which happen when processing progresses, for example when a ConnectorMessage is passed between Connector processing stages - see Checkpoints
Circuit Breaker	Circuit breakers protect systems from being bombarded with messages at a rate they would not be able to cope with.
Clearing and Settlement Mechanism (CSM)	A means by which a payment can be cleared and settled, typically it is an automated clearing house (ACH) but can also be used to refer to bilateral arrangements between banks.
Client Port	Provides a port/interface for client code to call an external or embedded IPF Business Function.
Client Request ID	The Client Request ID is a field that is reserved for allocation to a client-specific value that will allow association and inquiry to an IPF unit of work.
Collection	A grouping of documents in a NoSQL document database, broadly analogous to a table in a relational database. Typically, all documents in a collection have a similar or related purpose.
Completing	This defines whether the calling request should be considered completed when this response arrives. This is particularly useful when more than one response is expected during the interaction. For example, consider a request where the external system sends a technical acknowledgement followed by a final business response. In this case we would define two responses - one to represent the technical ack (non-completing) and one for the final business response (completing).
configurable enumerations	Some dynamic configurations on IPF support configurable enumerations. Client implementations can define their own enumeration values for some dynamic configuration attributes while also providing default enumeration values.
Connector framework	A connector is simply a set of common interfaces that provide a means for a flow to communicate with external systems.
Consumer	The user of an API.
Container	An immutable collection of software (application & supporting components, not operating system), packaged in a standardised format. Allows applications to be rapidly deployed in a repeatable and efficient way (since many containers can share a single operating system instance while remaining isolated).
Correlation	Establishing a link between messages, e.g. linking a PACS2 status report to a previously sent PACS8 payment.
Cosmos DB	A Microsoft document database that strives to be compatible with Mongo DB.
CQRS	Command Query Responsibility Segregation.
Creditor	The ISO20022 term for entities that receive funds as the result of a payment. Also known as the beneficiary or payee.
Cron	Cron is a time-based job scheduler used in Unix-like operating systems to automate repetitive tasks.
CRUD	Create Read Update Delete, an alternative approach to CQRS.
CS Agent Selection Settings	CS agent selection settings are the dynamic configurations defined by client implementations to provide a preference of which CSM Agents should be considered first for checking reachability of a payment. The clients can specify a selection order of CSM Agents based on currencies, payment types, service levels and settlement methods.
CT	An abbreviation for Credit Transfer.
Cucumber	Automated testing tool - Cucumber runs automated acceptance tests written in the behaviour-driven development (BDD) style, using a Given When Then syntax.
CVF	Credit Validation File in STEP2. It is a PACS2 giving a status update on a previously submitted PACS8.
Cypress	Cypress is a JavaScript-based testing framework designed for modern web applications, enabling developers and QA teams to automate GUI testing, including end-to-end (E2E), component, and integration tests. Unlike traditional tools like Selenium, Cypress runs directly in the browser, offering real-time execution and debugging.

DB	Abbreviation of database.
DD	Abbreviation of direct debit.
Dead letter / DLQ	A dead-letter queue (DLQ) is a secondary message queue that stores messages which fail to be processed in a primary queue, acting as a safety net for distributed systems. It isolates faulty or misdirected messages to prevent disruptions, enabling debugging and reprocessing.
Debtor	The ISO20022 term for entities that send funds as the result of a payment. Also known as the remitter or payer.
Decision	A decision allows us to perform some logic programmatically and then take different processing routes based on the outcome of that decision. For example, we may only want to run a fraud check if the value of the payment is over 50. In this case we can use a "Decision".
Decision Events	These event are used to drive the path a flow takes as the result of a decision.
Developer App	This is a lightweight simple example application that provides a view across the data generated by IPF. It's not a production utility, but is used to assist development.
Direct Debit Mandate	A mandate is an approval that gives the authority to process direct debits from an account for a given Creditor.
DN	A Distinguished Name (DN) is a unique identifier used in LDAP (Lightweight Directory Access Protocol) and X.500 directory services to locate and organize entries (such as users, groups, or devices) in a hierarchical directory structure.
Docker	Software packaging and deployment tool - Docker packages all elements of a software product (binaries, configurations, scripts, utilities etc.) into a single, lightweight container. This ensures that the software will always run the same, regardless of its environment. Because the Docker container is operating-system neutral, the packaged software can be deployed to different environments and on different flavours of Unix / Linux.
Domain event	Domain events are persisted facts about something which has occurred in your system. They cause a change of state in a flow.
Domain Functions	A domain function is a piece of logic within the domain of the flow that carries out processing useful to the flow. E.g. determine payment type.
Domain Specific Language	A Domain-Specific Language (DSL) is a specialized programming or scripting language designed for a specific application domain, unlike general-purpose languages (e.g., Python, Java). DSLs optimize efficiency and readability for narrow use cases by adopting syntax and abstractions tailored to their domain.
DVF	Debit Validation File In STEP2. It is a PACS2 giving a status update on a previously submittedPACS3.
Dynamic Processing Settings (DPS)	IPF Dynamic Processing Settings.
EBA	EBA Clearing is a provider of pan-European payment infrastructure such as STEP2 and RT1. It derives its name from the Euro Banking Association. Sometimes it is referred to as ABE or ABE-EBA which relate to the French version of its name.
Egress	Egress Traffic refers to data flowing out of a network or system to an external destination (e.g., the internet, another server, or a client device). It contrasts with ingress traffic, which involves incoming data.
EMF	Eclipse Modelling Framework: a language modelling format (similar to MPS) to allow the definition of Meta-Models.
Endpoint	An endpoint is any device or software component that connects to a network, serving as an entry or exit point for data communication. It enables interactions between systems, users, or services.
Enum	An enum (short for enumeration) is a custom data type that represents a fixed set of named constants, making code more readable and maintainable by replacing "magic numbers" or strings with meaningful labels.
EOD	End of Day.
E-Repo	ISO20022 E-Repository: a 100Mb binary file containing all the ISO20022 data in the EMF Meta-Model format.

ESB	Event Sourced Behaviour (or also an Enterprise Service Bus).
EuroSIC	Swiss (RTGS) system for euro payments between banks within Switzerland, operated by SIX Interbank Clearing Ltd.
Eurosystem	The monetary authority of the Euro area. The www.ecb.europa.eu/ecb/orga/escb/eurosystem-mission/html/index.en.htmlEurosystem] comprises the European Central Bank and the national central banks of the Member States whose currency is the Euro.
Event Behaviour	This is the behaviour that specifies what actions should be taken on receipt of an event. It is the core logic of a flow, e.g. when in State A if domain event 2 is received then do action X and move to State B.
Event Criteria	Used to control movement between states in a flow as part of Event Behaviour. On - this movement will happen upon the arrival of a single event (e.g. we may transition when receiving "Event 1") On any of - this movement will happen upon the arrival of one of multiple events (e.g. we may transition when receiving either of "Event 1" or "Event 2") On all of - this movement will only occur upon the arrival of multiple events (e.g. we may transition only after receiving both "Event 1" and "Event 2")
Event sourced	Event Sourcing is an architectural pattern where changes to an application's state are stored as an immutable, append-only sequence of events (facts) rather than overwriting the current state. Each event represents a discrete action (e.g., OrderPlaced, PaymentReceived) and includes contextual metadata (timestamp, user ID, etc.).
Eventually consistent	Eventual consistency is a data consistency model in distributed systems where updates to a data item propagate asynchronously across replicas, guaranteeing that given enough time without new changes all nodes will converge to the same state. It prioritizes availability and partition tolerance over immediate consistency.
External Domain	Represents some business domain - not our current flow's - that we need to interact with. For example, let's assume that we need to talk to a sanctions system during part of the flow. To support this, we would model that sanctions system as an external domain.
External message	A message exchanged between a flow and an external domain.
Fan-out/Fan-in	The Fan-Out/Fan-In pattern is a concurrency model in distributed systems that parallelizes task processing for efficiency, then consolidates results. Key Concepts Fan-Out: Distributes tasks across multiple workers (threads, processes, or services) for parallel execution. Fan-In: Aggregates results from all workers into a consolidated output.
Fednow	An American instant payment scheme.
Flo-Lang	One of the domain specific languages in IPF, it is used to define orchestration flows.
Flo-lang solution	This is effectively an overall project. A solution can contain many models.
Flow	The flow is the foundation of an IPF application, its the glue holding together the orchestration steps.
Flow Designer	The IPF product name for the Jetbrains MPS editor that is used to define a Flow. Flow Designer is included in the IPF core licence.
Freemarker	FreeMarker is a Java-based template engine for generating text output (HTML, emails, configuration files, source code, etc.) by merging dynamic data with pre-defined templates. It uses the FreeMarker Template Language (FTL), a lightweight, domain-specific language designed for presentation logic.
FX	Foreign eXchange, e.g. converting USD to GBP.
Generation	Converting a model specified in an IPF domain specific language into executable code.
Gherkin	Gherkin is a domain-specific language (DSL) used in Behaviour-Driven Development (BDD) to write executable specifications in plain, human-readable text. It bridges the gap between technical and non-technical stakeholders by describing software behaviour without implementation details.

Git	Software version control system - Open-source, distributed, software version control system. Used in IPF to perform version control across the IPF development environments.
Global state	A higher level state that the states found in flows.
Grafana	Grafana is an open-source observability and data visualization platform designed for real-time monitoring, analytics, and alerting. It enables users to query, visualize, and correlate metrics, logs, and traces from multiple sources through interactive dashboards.
Graphviz	Graphviz is an open-source tool for visualizing graphs and networks from structured data. It uses a declarative DOT language to define nodes, edges, and their relationships, which it then renders into diagrams (e.g., flowcharts, hierarchies, network topologies).
HMAC	HMAC (Hash-Based Message Authentication Code) is a cryptographic mechanism used in many places, such as within the SWIFT network to verify the integrity and authenticity of financial messages and within the IPF GUI to sign JWTs. It combines a secret key with a hash function (e.g., SHA-256) to generate a unique signature for each message.
HOCON	Human-Optimized Config Object Notation. A human-friendly configuration file syntax, used to define IPF configuration.
HTM	Human Task Manager. A component of IPF that allows flows to request humans to execute tasks and optionally return the result to the flow.
IBAN	International Bank Account Number. For more background see here.
IBAN Plus	IBAN Plus is a SWIFT-managed data product that enables financial institutions to validate IBANs, derive associated BICs (Bank Identifier Codes), and ensure compliance with cross-border payment standards (e.g., SEPA). It is part of SWIFT's reference data suite, alongside products like SEPA Plus and Payments Plus.
IBAN Structure	The IBAN Structure file (IBANSTRUCTURE_FULL_[YEAR].txt) is a SWIFT-provided resource that defines the technical format of IBANs for each country, ensuring compliance with ISO 13616. It is used by banks and financial software to validate and parse IBANs correctly. It is part of IBAN Plus.
ICF	An Input Credit File in EBA's STEP2. It is a PACS8.
IDE	Integrated Development Environment. E.g. Eclipse or IntelliJ.
Idempotency	Idempotency ensures that performing the same operation multiple times (e.g., due to retries or network issues) produces the same result as a single execution. It is critical for preventing duplicate transactions, erroneous charges, or inconsistent states in financial systems.
Identity Resolution	A feature of IPF that allows comparisons of names and addresses.
IDF	Input Debit file in STEP2. It is a PACS3.
Immutable object	Immutability is the property of data or objects that prevents modification after creation. Instead of altering existing data, operations generate new instances, ensuring consistency, security, and traceability.
Industry Data	Industry data are dynamic configurations that are maintained using data provided by industry sources through files. Examples of industry data are CSM Participants (populated using membership files from CSMs), Party Entity Directory and IBAN Plus directory (populated from SWIFTRef files).
Ingest	Data ingestion is the process of collecting, importing, and processing raw data from various sources into a storage or processing system (e.g., databases, data lakes, or streaming platforms). It is the first step in building data pipelines and analytics workflows.
Ingress	Ingress refers to the process of data entering a system from external sources. It is the first step in data pipelines, enabling collection and processing for analytics, storage, or real-time applications.
Initiation Behaviour	This is a behaviour that specifies the behaviour for each input that starts a flow.
Input Behaviour	Converts an input (an instruction or a response) combined with a response code to a domain event. E.g. Accounting system responded with code 404 > Domain Event "Account not Found".

Input Enricher	An Input Enricher generates (or updates) business data elements to be stored on a received event. The key point to understand in the difference to the "aggregate" type of mapping is that this data will be added to the event and persisted. This means for example that it will be sent and made available within the processing data stream for use outside of IPF.
Instruction	Instructions are the simplest thing flows receive from an external domain. It can be triggered by the external domain at any time and we will start processing. This can be thought of as the external domain pushing information to us.
IntelliJ	An Integrated Development Environment.
Intra Entity Party	An entity that shares ledger with the processing entity. Intra entity parties are usually branches of the processing entity.
Intra-entity payment	A payment within one ledger, often referred to as an 'on us' payment.
IPF	Icon Payments Framework.
IPF ISO20022 Model	A Java representation of the IS20022 Model.
IPF Mapping Framework	The IPF Mapping Framework translates between the flow data model and that of external systems in a configuration-driven, low-code way.
IPF Scaffolder	The IPF scaffolder is a java module that provides engineers with a quick and efficient way to bootstrap a new project using the IPF SDK. It can be executed via maven to generate a new IPF project that includes all the necessary dependencies and configuration to be able to run as a standalone project.
IQF	IQF is used in Step2 SCT, and is defined as "Input Inquiry File". Can contain camt.027, camt.087, camt.029 and pacs.028.
ISO 20022	ISO 20022 Financial Services - Universal financial industry message scheme. A standard for electronic data interchange between financial institutions, published by the International Organization for Standardization. ISO 20022 defines a common platform for the development of financial messages.
ISO 3166	ISO3166 is a standard that defines codes for the names of countries, dependent territories, and special areas of geographical interest.
ISO 4217	This standard establishes internationally recognized codes for the representation of currencies that enable clarity and reduce errors.
ISO20022 Meta-Model	Concepts, rules, types and relationships that formally map the interaction for Financial Message exchange.
Jackson	JSON/XML processor - Open-source Java-based library used to serialize or map Java objects to JSON and vice versa.
JaCoCo	Java Code Coverage library. Used as part of internal quality assurance. Provides analysis of unit test coverage in Development environment.
JAVA	Programming language used extensively in IPF.
Java Development Kit (JDK)	A software development environment used for developing Java applications.
Java Virtual Machine (JVM)	Virtual machine environment under which IPF runs.
JAXB	JAXB (Java Architecture for XML Binding) is a standardized framework for mapping Java objects to XML (marshalling) and XML to Java objects (unmarshalling). It simplifies XML handling by automating conversions, eliminating manual DOM/SAX parsing.
Jenkins	Build server - Located on the server where the project's main build is created, Jenkins triggers a new build every time a user checks in changes to the source code. This supports the process of continuous integration for testing and development. Icon use Jenkins internally for managing IPF builds.
Jest	Jest is a zero-configuration, batteries-included JavaScript testing framework developed by Facebook. It excels at unit, integration, and snapshot testing for frontend (React, Vue) and backend (Node.js) JavaScript applications.
JetBrains MPS	The tool used to develop IPF's domain specific languages and to use the languages to build solutions.

Jitter	In the context of retrying a failed network transmission, jitter refers to intentionally adding a small, random delay before retrying the request. This randomness helps prevent multiple clients from retrying simultaneously, which could overwhelm the server again.
JMS	Java Message Service (JMS) is a Java API that enables applications to create, send, receive, and read messages, facilitating loosely coupled, reliable, and asynchronous communication between components of a distributed system.
Journal	A journal is like a digital logbook that keeps track of important actions or changes made within a system or application. It records events, transactions, or updates to help developers understand what happened and when. Think of it as a way to keep everything organized and traceable, so if something goes wrong, it's easier to figure out why.
JSON	JSON (JavaScript Object Notation) is a simple text-based format for storing and exchanging data. It organizes information in key-value pairs (like "name": "John") and supports basic data types such as strings, numbers, arrays, and objects. JSON is widely used because it's easy for both humans to read and machines to process.
JUnit	JUnit is a free, widely used testing framework for Java applications, designed to help developers write and run automated tests for small, isolated pieces of code (like individual methods or classes). It ensures each component works correctly before integrating it into larger systems.
JWT	JWT (JSON Web Token) is a compact, secure way to transmit information between systems as a JSON object. It's commonly used for authentication and data exchange in web applications.
K8s	Kubernetes (because there are 8 letters between the K and the S).
Kafka	Kafka is a distributed, open-source platform designed for handling real-time data streams at scale. It acts as a high-throughput, fault-tolerant system that ingests, stores, and processes continuous data flows (e.g., logs, transactions, or sensor data) from multiple sources.
Kubernetes	Kubernetes (K8s) is an open-source container orchestration platform that automates the deployment, scaling, and management of containerized applications. Developed by Google and now maintained by the Cloud Native Computing Foundation (CNCF), it groups containers into logical units (like pods) for easy coordination across clusters of machines.
LAC	Liquidity Adjustment Checkpoint (SEPA) / Cycle (SEPAInst): a period during which liquidity transfers can take place
LAU	LAU Keys are a security mechanism used in systems like SWIFT to ensure the authenticity and integrity of messages exchanged between parties. They work by generating a unique digital signature, called a Local Message Authentication Code (LMAC), for each message. This signature is created using a pre-shared secret key and a secure algorithm (e.g., HMAC-SHA256). The receiving system verifies the signature to confirm that the message has not been tampered with and comes from a trusted source.
LDAP	Lightweight Directory Access Protocol is a standardized protocol used to access and manage directory services over a TCP/IP network. A directory service acts like an organized "phonebook" for network resources, storing hierarchical information about users, devices, roles, permissions, and other organizational data. LDAP simplifies the process of querying, updating, and authenticating against these directories.
Lightbend	Lightbend (now operating as Akka) is a software company originally founded in 2011 by Martin Odersky (creator of Scala), Jonas BonÃ©r (creator of Akka), and Paul Phillips. Headquartered in San Francisco, it specializes in tools for building reactive, distributed, and cloud-native applications on the Java Virtual Machine (JVM).
Low-code	Low-code is a software development approach that minimizes manual coding by using visual tools (like drag-and-drop interfaces) and prebuilt components to create applications. It accelerates development by abstracting complex code into reusable modules, enabling both professional developers and non-technical users ("citizen developers") to build functional apps faster than traditional methods.
Mapping Function	A mapping function is a piece of logic that is used to transform business data elements into different business data elements. They can be used in one to one, one to many or many to many examples.
Maven	Maven is an open-source build automation and project management tool primarily used for Java applications. Developed by the Apache Software Foundation, it standardizes the build process by automating tasks like compilation, dependency management, testing, and packaging.
Maven plugin	Maven Plugin is a modular component in Apache Maven that extends its core functionality by executing specific build tasks (called goals). Plugins enable reusable automation for tasks like compiling code, packaging artifacts, running tests, or generating documentation.
MDR	Message Definition Report: reference documents that are published by the ISO20022 Registration Authority.
MDS	Message Data Structure, a data structure for use within flows based on the ISO20022 message definitions.

Message Components	Message Definitions are comprised of Message Components.
Message Definition Identifier	ISO20022 identifiers, e.g. pacs.008.001.08.
Message Definitions	ISO20022 message definitions, e.g. FIToFICustomerCreditTransferV08.
Message log	A log of all messages sent and received by an IPF solution.
Message Schema	A message schema is a structured blueprint that defines the format, structure, and rules for data exchanged between systems. It ensures that both the sender and receiver understand and interpret the data consistently. Message schemas are commonly used in formats like JSON and XML.
Metrics Processor	The IPF Metrics Processor exists to assist in answering questions about IPF payment processing such as "How many payments have finished in the last hour?".
Modal	Modal refers to a GUI window or dialog that temporarily blocks interaction with the rest of the application until the user completes a required action or dismisses it. It forces focused attention by disabling the parent window's functionality, often dimming the background for visual emphasis.
MongoDB	Document-oriented database. MongoDB's document storage model is ideally suited for ISO20022 XML-style messages used for instant payments. It offers fast access and data retrieval for online transaction processing. IPF uses MongoDB for Transaction logging and enquiry, Message logging and enquiry, operational data such as States of endpoints, scheme participant availability, scheme specific data (for example, settlement cycles), Reference data, such as sorting code and participant directories.
Mutable object	Mutable Object is a programming concept referring to an object whose internal state or data can be modified after its creation. Unlike immutable objects, which remain constant once instantiated, mutable objects allow alterations to their properties, values, or structure without creating a new instance.
Nack	Negative ACKnowledgement. Also sometimes referred to as a NAK or a NOK (Not OK).
Namematcher	A Netowl product that is used in Identity Resolution.
NCCs	National Clearing Codes. E.g. in the UK, sort codes.
Netowl	The company that produces Namematcher, the software used in IPF's Identity Resolution.
Nexus Repository Manager	Nexus (Sonatype Nexus Repository Manager) is an enterprise-grade repository management platform designed to centralize, secure, and streamline the storage, distribution, and governance of software artifacts (e.g., binaries, libraries, containers) across the development lifecycle.
Node	A single application instance which is configured within a cluster of multiple instances.
Notification	Notification are the opposite of instructions in the flo DSL. These are used when we want to push our data out to an external domain.
Oauth	OAuth (Open Authorization) is an open-standard framework for secure, token-based authorization that allows third-party applications to access user data on another service (e.g., Google, Facebook) without exposing the user's credentials.
On Us Payment	A payment where both the debtor account and the creditor account are held with the same financial institution (or within the same group).
Open API	De-facto standard for API documentation.
Operational Data Store (ODS)	Operational Data Store.
OQF	OQF is used in Step2 SCT, and is defined as "Output Inquiry File". See also IQF - Input Query File.
Original reason code	The original (e.g. that of an external domain) reason for something. E.g. account closed, insufficient balance.
Original reason text	The original (e.g. that of an external domain) explanatory text relating to the result of an action.

Original response code	The original (e.g. that of an external domain) status of something. E.g. accepted, rejected, blocked.
Originating Bank	The bank that originates (sends) a message (Outbound Customer Credit Transfer). In ISO20022 terms, for a credit transfer it's the Debtor Bank and in direct debit it's the Creditor Bank.
Orika	Orika is a Java-based bean mapping framework that automates the copying of data between objects with different structures (e.g., DTOs, domain models, API responses). It eliminates manual boilerplate code by dynamically generating optimized mappers at runtime using bytecode generation, offering performance close to hand-written mapping.
Orphan	An unexpected message e.g. getting a clear and settle response without having sent a request, or getting a timeout status report without having received the payment that is being timed out.
PACS	PAmyent Clearing and Settlement messages from ISO20022, e.g. pacs.008 is an interbank customer credit transfer, pacs.003 is an interbank direct debit.
PAIN	PAyment Inititation messages from ISO20022, e.g. pain.001 is a customer credit transfer initiation.
PART	An ISO20022 group payment status that means the file was partially accepted.
Participant	A member (generally a bank) of a payment scheme/Automated Clearing House.
Party entity directory	Party Entity Directory is a dynamic configuration that can be used to look up Party Entities to retrieve the party entity details like identifiers and address details. Party entity directory can be populated with varios industry data sources as well as custom client specific source of reference data. Party entity data retrieved from party entity directory can be used for payment enrichment purposes.
Passivation	A flow passivates when it moves to a specified state and the Action from the Event Behaviour table is executed. Passivating essentially removes the flow entity from memory, to be rehydrated (i.e. reloaded) and activated at a later point, e.g. future dated payments or lengthy sanctions investigations.
Payment journey	IPF and ODS will refer to a collection of processing flows as a Journey Type. The currently available Journey types are as follows: PAYMENT > Pacs.008, Pain.001, Pacs.002, Camt.054 based RECALL > Camt.056, Pacs.004, Camt.029 BULK > Pain.001 group header or Pacs.008 group header BATCH > Pain.001 Payment Information
Payment Scheme	A payment scheme is a set of rules which have agreed upon to execute transactions through a specific payment instrument (such as credit transfer, direct debit, card, etc). It is different from a payment system, which is a technical infrastructure that processes transactions in line with the rules defined in a payment scheme. Generally there is one payment system that implements the scheme, but not always, most notably in the case of SEPA.
Payment Services Provider (PSP)	Payment Services Provider - this is the entity which owns and runs the Payments Processing capability (aka Payments Processor). This might not be a bank or a Financial Institution, it could be run by another entity.
Payment Status Report (PSR)	The ISO20022 "FI to FI Payment Status Report" message (pacs.002), which is principally used to respond (accept, reject, etc.) to a payment request.
Payment Type	In IPF a payment type is a classifier for payments that is completely determined by the user of IPF.
PDS	Processing Data Structure. Allows users of IPF to create completely bespoke data structures, which are often based on theISO20022 business model, but do not have to be.
Pod	A group of one or more co-located containers, deployed together as a unit, with shared storage, shared network resources, and run in a shared context.
Pojo	POJO (Plain Old Java Object) is a term for a simple Java class that adheres to basic Java conventions without being tied to specific frameworks, interfaces, or annotations. It emphasizes minimalism and reusability by avoiding external dependencies.
Port	An interface to provide entry and exit point to the core application, agnostic of the specific external connection implementation.

Processing context	IPF's Processing Context is an object which contains a collection of unique identifiers related to IPF processing (including UowId, AssocationId & ClientRequestId)
Processing entity	An organisational subdivision within a deployment of IPF. E.g. a client may have a branch in Germany and a branch in France. Each branch is a different processing entity and so can have different configuration, user access etc.
Processing Settings	Processing settings are dynamic configurations that are specific to a client implementation and maintained by client teams using IPF DPS capability. Examples of processing settings are Processing Entity, Generic Processing Settings, Agent Settings, Agent Clearing & Settlement Settings, CS Agent Selection Settings.
Producer	In a reactive system, a producer (or Publisher) is a component that emits data streams (events, messages, or values) asynchronously to one or more consumers (Subscribers). It adheres to the Reactive Streams specification, enabling non-blocking, event-driven data flow with backpressure control.
Prometheus	Prometheus is an open-source monitoring and alerting toolkit designed for cloud-native and dynamic environments (e.g., Kubernetes, microservices). It collects, stores, and analyses time-series metrics numeric data points recorded with timestamps enabling real-time performance tracking and issue detection.
PromQL	PromQL (Prometheus Query Language) is the functional query language used in Prometheus to select, aggregate, and analyse time-series metrics in real time. Designed for monitoring dynamic systems, it enables users to extract insights, create alerts, and visualize data through expressive syntax tailored for temporal data.
QVF	A Query Validation File, in response to an Input Query File (IQF) in EBA STEP2. It is aPACS2.
R transactions / R messages	Messages relating to a payment, e.g. Request to cancel, Return, Refund etc.
RA	The ISO20022 Registration Authority.
Reachability	Participants of a scheme/ACH are reachable. This is the reachability of the scheme.
Reactive	Reactive Programming is a declarative programming paradigm cantered on asynchronous data streams and automatic propagation of change. It enables applications to dynamically respond to real-time events (e.g., user inputs, sensor data) by modelling data flows as composable, event-driven sequences.
Reactive streams	Reactive Streams is a standardized asynchronous stream processing specification designed for handling data flows with non-blocking backpressure ensuring that fast producers don't overwhelm slow consumers.
Read side	Read Side (in Event Sourcing & CQRS) refers to the part of a system optimized for querying data, where materialized views or projections are generated from an event log to serve fast, structured responses to clients. Unlike the write side (which handles commands and stores events), the read side focuses solely on efficient data retrieval.
Reason code	A "Reason Code" is a reason why the response code is set. So for example your response code could be "Rejected" with a reason "Incorrect Currency". In ISO 20022 this is a status reason.
Reason text	The IPF implementation (NOT that of an external domain) explanatory text relating to the result of an action.
Recall	A request to return funds previously sent, typically uses aCAMT56.
Regional Brands	The EBA STEP2 term for Indirect Participants
Rehydrated	Rehydration is the process of rebuilding an object's current state by replaying a sequence of past events from an event log. It's a core concept in event-sourced systems, caches, and frontend state management.
Remember Entities	Remembered Entities (or Recoverable Entities) are objects in an event-sourced system that maintain their state by storing and replaying a sequence of past events. Unlike traditional databases that persist only the latest state, these entities "remember" their entire history
Request	A request is used when a flow needs a response back from the external domain in reply.

Request To Pay (R2P, RTP)	An additional value-added service offered by some instant payment schemes (including TCH in the USA, GIRO in Hungary). The Request To Pay is a message initiated by a party (business or individual) that wants to be paid, containing all relevant details of the payment required. The request to pay is sent from the requester's bank to the bank of the party being asked for payment, via the central infrastructure of the instant payment scheme. The receiver of the request can then simply review details and either accept or reject it. Accepting the request will trigger the sending of an instant payment using the details contained in the request to pay.
Requester	In a scheme that verifies Creditor names prior to payment initiation (e.g. Confirmation of Payee in the UK or Verification of Payee for SEPA), the requestor is the organisation from where the payment will be initiated should the identity be confirmed.
Resolution of investigation (ROI)	A resolution of investigation in payment processing is typically used to respond to a recall and uses aCAMT29. Usage varies by scheme, e.g. in SEPA ROI is only used for a negative answer for a recall, whereas in TCH RTP it used for negative and positive.
Responder	In a scheme that verifies Creditor names prior to payment initiation (e.g. Confirmation of Payee in the UK or Verification of Payee for SEPA), the responder is the organisation that holds the correct name information.
Response code	A "Response Code" is an expected outcome code for a response that could be used for onward processing. E.g. Rejected, Accepted, Posted, Blocked etc. In ISO 20022 this is a Status.
REST	REpresentational State Transfer. REST is an architectural style for designing networked applications, emphasizing simplicity, scalability, and stateless communication over HTTP. It standardizes how systems expose and interact with resources (data or services) using uniform interfaces.
Return	Typically a PACS4, this returns funds previously received to the originator of the payment. In some schemes the recipient of the funds can initiate a return, in some schemes a return can only be sent when the originator has requested so via a recall.
RJCT	An ISO20022 payment status - Rejected.
RRR (See also R transactions / R messages above)	Recalls, Returns and Results Of Investigations.
RSF	Result of Settlement File. An STEP2-standard file for payment instructions/returns that have been cancelled or revoked and are returned by the STEP2 Central System to the Direct Participant who was the Sender of the ICF in which the cancelled instructions were delivered to the Central System.
RT1	EBA Clearing's clearing and settlement infrastructure for SCT Inst. Currently banks and other PSPs can connect to RT1 via SIAnet or EBICS.
RU	Request Unit (RU) is the performance currency of Azure Cosmos DB, representing the computational resources (CPU, memory, IOPs) required to execute database operations like reads, writes, queries, and updates. It abstracts infrastructure complexity into a single measurable unit for cost and capacity planning.
S3	A file storage mechanism offered as part of Amazon Web Services (AWS).
SAML	SAML (Security Assertion Markup Language) is an XML-based open standard for securely exchanging authentication and authorization data between parties typically between an Identity Provider (IdP) (e.g., Azure AD, Okta) and a Service Provider (SP) (e.g., SaaS apps like Salesforce). It enables Single Sign-On (SSO), allowing users to access multiple services with one set of credentials.
SCALA	Programming language - Object-oriented programming language (Scalable Language) in which Akka is written.
SCF	Settled Credit File. An EBA STEP2-standard file of payment instructions that have been successfully processed by CGS and are transmitted by the Central System to the receiving Direct Participant. It is a PACS8.
Scheduler	Scheduler ensures that the execution of payments is started at a time that allows any needed value added services (Future dated payments, Batch payments etc) to be applied and their value date time to be met. The scheduler monitors the payment data store for individual and bulk payments whose value date & time and associated value added services (Netting, Bulking, Enrichment, Remittance handling etc.) makes them eligible for execution. The scheduler also monitors the payment data store for any re-occurring payment schedules that require a payment to be created and executed in order for value added services to be applied and the value date & time schedule to be met. Whenever a payment is found or created that is eligible for execution the scheduler invokes any needed value added services and triggers the execution of the payment. Scheduler is usually used for make payments, e.g. 1. Future Dated Payments 2. SEPA CT Payments 3. SEPA DD etc.
Schematic	Schematics in Angular are like automated instructions or "recipes" that developers use to create or modify parts of a GUI project. They help streamline repetitive tasks, ensuring consistency and saving time.

Scheme Pack	Scheme Packs are a specific type of Additional Optional Module each related to the processing concerns for a specific CSM within a scheme. Each available Scheme Pack is licensed separately. A Scheme Pack can be developed by Icon or a client and includes the functionality that is required to process outgoing and incoming payments/transactions with respect to a specific CSM.
SCT Inst	SEPA Credit Transfer Instant Payments. Scheme layer of Euro instant payments. Based on SEPA Credit Transfer. The scheme is defined and managed by the European Payments Council and implemented by multiple CSMs.
SEPA	Single Euro Payment Area
SIC	Swiss Interbank Clearing.
Sinks	A data sink is the destination endpoint in a data flow system where processed or collected data is stored, forwarded, or consumed. It acts as the final repository or output for data after it passes through sources, transformations, or pipelines.
Six Bank Master	An industry directory listing participants in Swiss Interbank Clearing (SIC).SIX is the technology provider for SIC.
Snapshot	A snapshot is a static, point-in-time copy of a dataset or system state, used to preserve historical data, enable recovery, or analyse changes over time. It captures the exact state of data at a specific moment, often for auditing, backups, or incremental processing.
Sources	In stream processing, sources are the entry points that generate or emit real-time data streams. They feed raw data into a streaming pipeline for processing, analysis, or storage.
Split Brain	Split brain is a critical failure scenario in distributed systems where a cluster of nodes becomes partitioned due to network issues, causing subsets of nodes to operate independently without recognizing each other. This leads to data inconsistency and potential corruption as each partition makes conflicting decisions.
Split Brain Resolver	A split brain resolver is a mechanism in distributed systems designed to detect and recover from network partitions (split brain scenarios) by enforcing policies that maintain data consistency and cluster stability. It ensures that only one partition remains active or authoritative, preventing conflicting updates.
Splunk	Splunk is a data analytics platform designed to ingest, index, search, and visualize machine-generated data (logs, metrics, events) in real time. Its architecture is highly scalable, supporting deployments ranging from single-server instances to distributed, multi-site clusters.
Spring	Spring (and Spring Boot/Spring Cloud) is a Java-based framework for building distributed systems, offering tools to implement common patterns like microservices, event-driven architectures, and resilience mechanisms.
SSO	Single Sign On.
Standard Settlement Instructions (SSI)	These are details about standard banking relationships between banks by currency. Note: "SWIFT Ref SSI Plus" is a SWIFT supplied set of reference files. It is possible for customer preferences ("Correspondent Bank Preferences") to override those published by the Bank, as these only reflect the Banks agreements with other Banks. Where a customer (i.e. an agency bank) has other agreements in place, these may override the Banks agreements.
State	States represent a point in the processing of your system. Sometimes referred to as a DSL state to distinguish from Global States.
State Transition	When a flow moves from one state to another.
Status Request Scheduler (SRS)	The Status Request Scheduler (SRS) is a feature of the CSM Service that allows users to automate the sending of status request messages (typically PACS28s) after initiating certain payment actions.
STEP2	Straight Through Euro Processing 2. A batch based SEPA clearing system run by the EBA.
STET	STET is a provider of SEPA clearing services.
Subflow	A subflow is a reusable flow component. It is essentially the same as a flow in that it has states, input behaviours and event behaviours. However, a subflow has no life of it's own and is only used as a mechanism of reuse and therefore MUST be contained within an outer flow.

Swagger	Swagger is an open-source toolset for designing, documenting, and testing RESTful APIs based on the OpenAPI Specification (OAS). It standardizes API descriptions in machine-readable formats (YAML/JSON), enabling automation in documentation, client SDK generation, and testing.
SWIFT	The Society for Worldwide Interbank Financial Telecommunication A key part of global payments infrastructure.
SWIFT AGI	SWIFT Alliance Gateway Instant is a connectivity plugin or adapter used to integrate financial applications with SWIFT Alliance Gateway, a centralized messaging hub for secure and scalable communication across SWIFT networks (FIN, InterAct, FileAct).
Swift BICDir2018	BIC Directory 2018 is a simplified version of BIC Plus file that provides limited information on the BIC8 value, branch code, value added services and address information. BIC Directory 2018 will be replaced by BIC Directory in 2025.
Swift Exclusion List	The SWIFT exclusion list refers to financial institutions (typically banks) barred from accessing the SWIFT messaging network, often as part of international sanctions. This disrupts their ability to conduct cross-border transactions, trade, or access global financial markets.
System Event	These occur when something happens to the system. There are a set of default events but they can also be tailored to be specific to individual needs.
T2-RTGS	Target 2 Real Time Gross Settlement Service
TARGET Instant Payment Settlement service (TIPS)	TARGET Instant Payment Settlement service is a SEPA Inst CSM.
TCH	The Clearing House. Long-established US payments clearing and settlements provider,
TCK	A Technology Compatibility Kit (TCK) is a standardized test suite used to verify that software products (e.g., interfaces, applications) comply with specific technical requirements, ensuring interoperability within a defined ecosystem.
Technical Duplicate Check	Technical duplicate check for an inbound or outbound payments message using hash techniques or any other solution to ensure that a file is not duplicate. Technical duplicate check is intended to detect files/messages that are totally identical which are usually caused by malfunction application or middle ware. This service is often provided by the channel. Technical duplicate check is different from business duplicate check. If a payment is found to be a duplicate (with same payment reference, amount, sender and beneficiary etc) then the payment is rejected back to the originator.
Test-FW	Either the IPF test framework or the Icon Test Framework.
Throttling	Throttling (or rate limiting) is a technique to control request throughput in APIs, services, or networks preventing overload, ensuring fairness, and maintaining system stability.
Transport	A means of transferring data between systems, e.g. Kafka, MQ, http etc.
TTL	Time to Live. TTL is a database feature that automatically expires and deletes records after a specified duration or at a fixed timestamp. It optimizes storage, reduces costs, and complies with data retention policies.
Twelve-Factor App	The Twelve-Factor App methodology is a set of best practices for building modern, scalable, and maintainable software-as-a-service (SaaS) applications. Originally proposed by Heroku in 2011, it emphasizes portability, resilience, and automation in cloud-native environments.
UETR	Unique End-to-end Transaction Reference, universally unique identifier to provide an end-to-end reference of a payment transaction.
Unified Modelling Language (UML)	Unified Modelling Language.
Uniform Resource Identifier (URI)	Uniform Resource Identifier.
Unit of work ID (UOWID)	The default unique identifier for a transaction in an IPF solution.
Upsert	Upsert (a portmanteau of update + insert) is a database operation that automatically inserts a new record if it doesn't exist, or updates an existing one if it does. It eliminates the need for separate INSERT and UPDATE statements, improving efficiency and atomicity.

UUID	Universally Unique Identifier. Also known as GUID (Globally Unique Identifier).
Verification of Payee (VOP)	An IPF service that allows participants to verify the identity of a creditor account before their client initiates a payment to that account. Note it also the name of one of the schemes that the IPF function VOP supports, being the verification scheme for SEPA.
Write side	In Command Query Responsibility Segregation (CQRS), the write side (or command side) handles all data modifications (inserts, updates, deletes) and enforces business rules, while the read side serves queries. This separation optimizes performance, scalability, and domain logic clarity.
XJC	XML to Java Compiler is a command-line tool included in the Java Development Kit (JDK) as part of JAXB (Java Architecture for XML Binding). It compiles XML Schema (XSD) files into annotated Java classes, enabling seamless conversion between XML and Java objects.
XML	eXtensible Markup Language is a markup language designed for structuring, storing, and transporting data. Unlike HTML, which focuses on displaying data, XML defines custom tags to describe content in a machine and human readable format.
Xms	The initial amount of memory allocated.
Xmx	The maximum amount of memory that can be allocated.
XSD	XML Schema Definitions are a standard format for expressing a schema, XSDs are provided for each Message Definition by the ISO20022 Registration Authority.
XSL	eXtensible Stylesheet Language is a language for styling and transforming XML documents, primarily through XSLT (XSL Transformations). It consists of three parts: XSLT: Transforms XML into other formats (HTML, XML, text). XPath: Navigates XML nodes (used within XSLT). XSL-FO: Formats XML for print/PDF (now largely replaced by CSS).
YAML	YAML Ain't Markup Language is a human-readable data serialization format designed for configuration files and data exchange. It prioritizes simplicity and readability, using indentation and minimal syntax compared to XML or JSON.

\n

Frequently Asked Questions

1 pages in this subsection

Frequently Asked Questions

Source: https://docs.ipfdev.co.uk/reference/current/faq/faq.html

Frequently Asked Questions

This section provides answers to commonly asked questions, split down into several sections. You can use the search bar to search the whole site, but these FAQs should link to associated documentation within the developer docs.

> This is a living document and questions are being added as they become obvious.

General

- What is IPF?
- Is IPF cloud native?
- What is an AOM?
- Does IPF support ISO20022?
- Are there scheduled maintenance windows in payment solutions built using IPF?
- Can processing applications built with IPF support multiple bank legal entities?
- How can I learn IPF?
- How to identify a single transaction?
- Is IPF only for instant payments?
- Is IPF only for single individual payments?
- Does IPF support bespoke payment validation?
- Does IPF provide a user interface?
- Does IPF provide a user interface to manage payment returns, recall/reversal?
- What is a processing entity?
- What is Payments DSL?
- Does IPF provide a business workflow/process modeller?
- How do I define flows using IPF?
- What is a domain event?
- What is a system event?
- How can I view system events?
- As a bank can I define my own system events if not available in the default catalogue?
- How can I know what happened to my payment?
- Is it possible to see what raw messages exchanged for a payment?

Integration - Connector Framework

- Will any outgoing message be stored in IPF?
- What are the options to integrate a solution built with IPF to other bank systems?
- What integration points does IPF provide?

Additional Optional Modules (AOM)

- What is a Scheme Pack?
- What is the CSM Reachability Service?
- What is the Identity Resolution Service?
- What is the Payment Status Notification Service?
- What is the ODS?
- What is the Human Task Manager Service?

Business Functions

- Does IPF Perform technical and functional duplicate checking?
- What is Dynamic Processing Data?

Operational Data Store (ODS)

- What is the ODS?
- What are the main features provided by ODS?
- Is it possible to create downstream feeds for other bank applications or data store?
- What is the maximum number of transactions exportable from the ODS search?

Monitoring & Alerting

- What is the IPF offering regarding Business Alerting and Monitoring?
- Does IPF provide default set of metrics?

- What is message logging?
- What are logging options IPF supports?

Resilience

- How do solutions built with IPF cope with expected and unexpected failures?

Performance & Scalability

- What response times can be expected and are usual for solutions built using IPF? Do you have some benchmarks?
- How can a solution built with IPF be scaled to changing load or performance needs? What would be the impact on the underlying infrastructure ?

Deployment

- Are there any specific environment requirements that a bank would need to meet to host solutions built with IPF?
- What are the hosting options available for IPF solutions?
- Does IPF support Rolling Upgrades?

Development

- What are the tools and technologies required to start developing application using IPF?
- What version of Java is IPF using/supporting?
- What is the technology stack IPF is based on?
- What documentation exists for IPF?

Security

- Is IPF code base checked against known vulnerabilities?
- How is data at rest protected and which encryption options are available?
