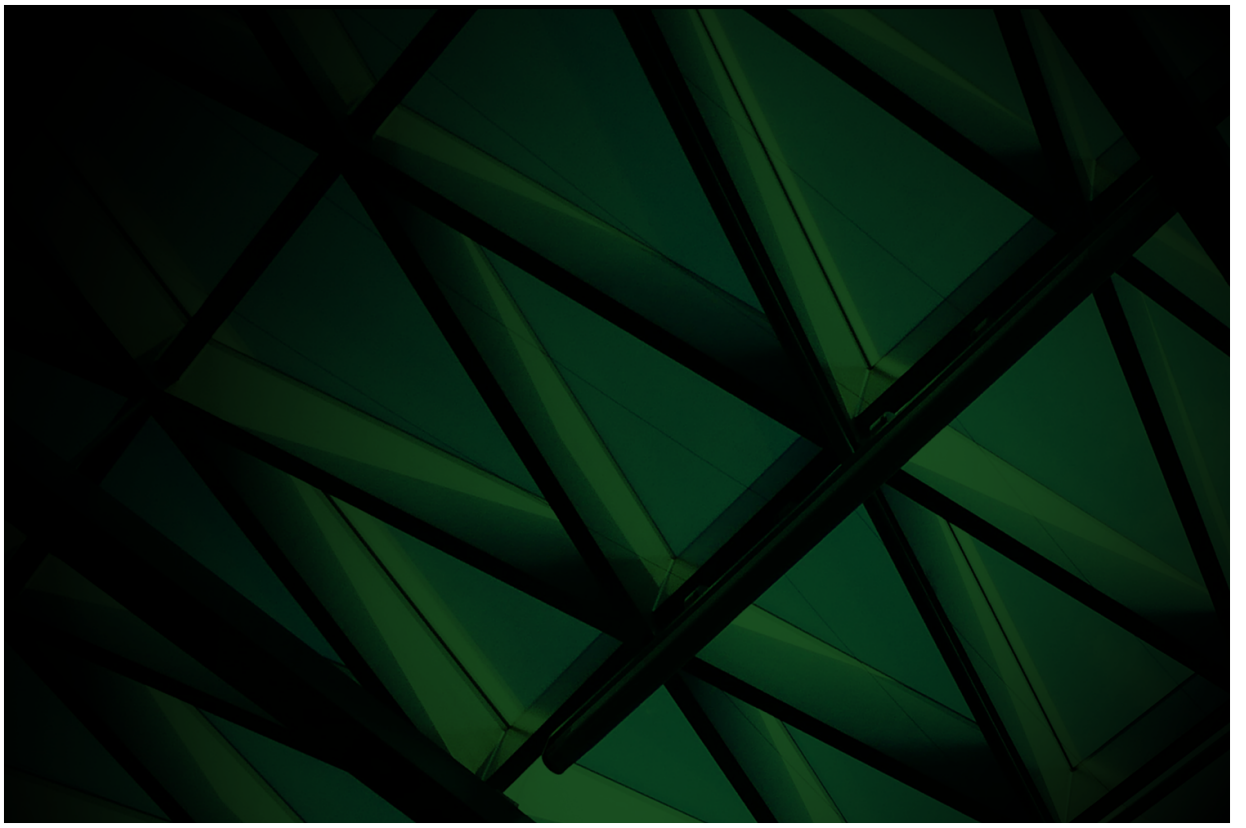


Jade Data Refinery Release Notes

VERSION R1

jade platform

Copyright©2026 Jade Software Corporation Limited. All rights reserved.

Jade Software Corporation Limited cannot accept any financial or other responsibilities that may be the result of your use of this information or software material, including direct, indirect, special or consequential damages, or loss of profits. There are no warranties extended or granted by this document or software material.

You should be very careful to ensure that the use of this software material and/or information complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without the express written permission of Jade Software Corporation Limited.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Copyright © 2026 Jade Software Corporation Limited.

All rights reserved.

JADE is a trademark of Jade Software Corporation Limited. All trade names referenced are the service mark, trademark, or registered trademark of the respective manufacturer.

For details about other licensing agreements for third-party products, you must read the Jade **Readme.txt** file.

Jade Data Refinery Release Notes

Jade Data Refinery (JDR) R1 is an event transformation and streaming solution for the Jade Platform that produces enriched, business-ready data streams for analytics, enterprise integration, and downstream application consumption.

JDR provides source-centric event enrichment by combining Change Capture on a dedicated refinery secondary database with declarative event mapping and distribution. Event mappings define the business data, referenced context, derived values, and metadata required by downstream consumers. The event mappings are compiled into versioned contracts and applied on the dedicated refinery secondary system.

JDR R1 provides the core capabilities required to:

- Define event mappings declaratively.
- Capture referenced context associated with the originating transaction.
- Add computed and derived values.
- Compile, validate, version, and activate event mapping contracts.
- Publish enriched Event-Carried State Transfer (ECST) events.
- Distribute events through the existing ESP pathways for Apache Kafka and Azure Event Hubs.

Notes The release numbering cycle of JDR is separate from the Jade Platform release numbering. Where relevant, the minimum version of Jade required for JDR is specified.

JDR R1 refers to the first general release of Jade Data Refinery.

JDR R1 requires Jade Platform 2025.0.03 (2025 R3) or higher.

JDR R1 Highlights

This section contains details about the core capabilities in JDR R1.

Enriched Event-Carried State Transfer Events

JDR supports the Event-Carried State Transfer (ECST) pattern, which allows event mappings to include the business state and referenced context required by downstream consumers.

An enriched event can include:

- Mapped business data from the changed object.
- Selected context from referenced objects.
- Derived and computed values.
- Standard runtime and event metadata.

Copyright 2026, Jade Software Corporation Limited All rights reserved

Enriched events reduce the need for downstream consumers to perform repeated lookups, joins, or consumer-specific enrichment.

ECST provides the following benefits.

- Consumers can receive the context they need as part of the event, which reduces consumer coupling.
- Referenced context is captured in association with the originating transaction, which provides temporal consistency.
- Captured context remains associated with the event when the event is processed or replayed downstream, which provides replay consistency.
- Multiple downstream systems can consume enriched business events without recreating the same enrichment logic.

Reference Context Capture

Reference Context Capture (RCC) captures referenced and derived context while transactions are replayed on a dedicated refinery secondary system. RCC preserves context associated with the originating transaction while keeping enrichment processing off the Online Transaction Processing (OLTP) commit path.

RCC can capture:

- Natural keys and selected scalar attributes from referenced objects.
- Derived and computed values declared by the event mapping.
- Small embedded aggregates required by the event contract.

Without RCC, equivalent enrichment could require downstream joins, replicated reference datasets, on-demand lookups, or consumer-specific transformation pipelines.

Declarative Event Mapping Specifications

JDR event mappings are authored as JSON Event Mapping Specifications (EMS). This document refers to an Event Mapping Specification as a *mapping specification*.

A mapping specification defines:

- The Jade class changes that produce events.
- The target event name.
- The fields included in the event payload.
- Field names and target shapes.
- Approved type conversions.
- System metadata fields.
- Computed and derived fields.

The JDR mapping tools compile a mapping specification into an Event Mapping Contract (EMC). This document refers to an Event Mapping Contract as a *mapping contract*. This is the validated runtime mapping contract used by the JDR Change Capture Process (JDR CCP) and the JDR Event Stream Producer (JDR ESP). Mapping contract files are generated artefacts and are not intended to be edited manually.

Separating the authoring mapping specification from the runtime mapping contract ensures that the mapping reviewed and validated before deployment is the same mapping used at run time.

Copyright 2026, Jade Software Corporation Limited All rights reserved

Versioned Mapping Bundles

Event mapping artefacts are packaged as Event Mapping Bundles (EMB). This document refers to an Event Mapping Bundle as a *bundle*.

A bundle is the unit of deployment and activation. It contains one or more mapping contract files, generated JSON Schema artefacts, and a bundle manifest. A bundle is eligible for activation only after it has been validated and resolved on the refinery secondary for the schema context against which the JDR CCP and EVR will run.

Note A bundle does not contain any active mapping contracts until after it is activated.

The steps in a bundle lifecycle are described in the following list.

1. Author the transaction mapping specification files that describe the required mapping contracts.
2. Compile mapping specification files into mapping bundles.
3. Validate and resolve the mapping bundle against the deployed Jade logical data model.
4. Activate the validated mapping bundle through the JDR CCP, to generate active mapping contracts.

After mapping bundle activation, the EMCs in the bundle become the active mapping contracts used by change capture and the JDR ESP.

Mapping changes take effect only after a new bundle is activated, which provides downstream consumers with an explicit, versioned contract lifecycle.

Computed and Derived Fields

JDR mappings can include values calculated by Jade instance methods marked with the **refinery** method option.

Refinery methods can provide:

- Derived business values.
- Calculated fields.
- Aggregated values.
- Values that are not represented directly by a persistent property.

Refinery methods are subject to additional restrictions that ensure they are non-updating and suitable for execution during refinery processing.

Computed fields participate in the same mapping model as ordinary properties and can be renamed, schema-typed, and transformed.

System Fields and System Bindings

JDR exposes standard platform and runtime metadata as virtual, read-only system fields that are available to event mappings.

JDR R1 system fields include:

- eventId
- schemaId
- eventType
- eventTimezoneOffset

Copyright 2026, Jade Software Corporation Limited All rights reserved

- eventSequenceNumber
- contentType
- transactionId
- objectType
- eventTime

System bindings provide a centralised way to define naming conventions for system-provided values across mappings in a bundle, which reduces repeated mapping configuration and helps maintain consistent event contracts.

Standard ESP Compatibility

JDR R1 builds on the existing standard Event Stream Producer (standard ESP) change capture and distribution infrastructure.

Standard ESP operation remains supported. Sites that do not use JDR mapping bundles can continue to run standard ESP independently of JDR.

JDR retains the standard ESP event envelope while allowing the mapping contract to define the enriched event payload.

Supported distribution targets are:

- Apache Kafka.
- Azure Event Hubs using native Advanced Message Queuing Protocol (AMQP).
- Azure Event Hubs using the Kafka surface.
- File sender.

Platform and Deployment Requirements

JDR R1 requires:

- Jade Platform 2025.0.03 (2025 R3) or higher.
- A dedicated refinery secondary system.
- A valid JDR licence.

Contact Jade licensing at licensing@jadeplatform.com to discuss your licensing requirements.

JDR R1 is supported on the operating systems supported for the applicable Jade Platform release.

The refinery secondary system isolates JDR Change Capture and enrichment processing from the primary Online Transaction Processing (OLTP) workload.

Limitations and Restrictions

The following capabilities are currently not supported in JDR.

- Delta-format events.
- Field-specific large data handling policies.

Copyright 2026, Jade Software Corporation Limited All rights reserved

- Advanced schema management capabilities beyond the R1 feature set.
- Advanced error handling policies beyond the R1 feature set.
- A dedicated mapping authoring user interface. Mapping specification authoring is file- and tool-based.
- Transforms outside the supported transform set.
- Historical replay from a retained enriched event log.
- Automatic activation of a new mapping bundle following a Jade schema migration.
- Standard structured error-envelope emission for capture, transform, and serialisation failures.

The following platform restrictions also apply.

- A secondary database cannot be configured with both the **RPS relational** subrole and the **Refinery** subrole.
- Primitive array updates are not captured.
- Collection updates are not captured directly. Where configured, relevant changes can instead be published through participating inverse objects.
- Delete events are not fully supported on an encrypted database. When the database is encrypted, delete events contain only event metadata and the Object Identifier (OID).

Upgrade and Coexistence

JDR R1 does not require you to replace an existing standard Event Stream Producer (standard ESP) deployment.

Standard ESP and JDR can operate side by side using separate capture environments.

Supported configurations include:

- Standard ESP running on the primary database, with JDR running on a dedicated refinery secondary system.
- Standard ESP running on a native SDS secondary database, with JDR running on a separate dedicated refinery secondary system.

These configurations allow sites to introduce JDR incrementally while retaining existing standard ESP integrations where required.

Copyright 2026, Jade Software Corporation Limited All rights reserved