

InterSystems IRIS data platform 2025.1 EMリリース 新機能と改善点について

2025年4月24日(木) 13:30-14:15
インターフェースシステムズジャパン株式会社
セールスエンジニア 奥山 朋 (おくやま とも)



IRIS data platform 2025.1 EMリリース：新機能と改善点について



ウェビナーの概要

このウェビナーでは、InterSystems IRIS data platform 2025.1 EM（Extended Maintenance）リリースに搭載される新機能と改善点を包括的に紹介します。EMリリースは長期サポートが提供され、安定性が重視されるため、多くのお客様が本番環境で採用されるバージョンです。2024.1 EMリリース以降の機能強化も含め、技術者の皆様が新バージョンを最大限に活用できるよう情報を届けします。

本ウェビナーでは、製品の新機能や改善点について、実際の実行環境での画面を共有しながら解説します。理論的な説明だけでなく、具体的なコード例や実際の動作デモンストレーションを通して、新機能の実用的な活用方法を分かりやすく紹介します。これにより、参加者の皆様はすぐに実践できる知識を得ることができます。

対象者：

IRIS data platform開発者

データベース管理者

前提知識：

IRIS data platformの基本的な知識

過去のバージョン（特に2024.1以降）の使用経験があると望ましい

ウェビナー終了後に準備出来次第YouTubeに本ウェビナー動画を公開いたしますので、ご同僚の方にもご案内ください。



InterSystems IRIS 2025.1機能強化 まとめ

主要な機能強化ポイントまとめ

大幅なパフォーマンス向上

データベース拡張処理が最大100倍高速に
ストレージ効率の大幅改善
分析処理の高速化

- システム停止やパフォーマンス低下の心配なく、ビジネスの成長に合わせてスケール
- 圧縮アルゴリズムの最適化により大容量データを効率的に管理し、コスト削減を実現
- 複雑なクエリも瞬時に結果を返し、意思決定のスピードを加速

AIをビジネスの核心に

誰でも容易にAIを活用可能に
ベクター検索の大幅なパフォーマンス向上
LLMとの連携が容易に

- 専門知識不要で既存データからAIソリューションを構築
- 大規模データセットでも瞬時に類似検索を実行
- 組織固有のナレッジをAIに取り込み、より精度の高い回答を実現

運用の簡素化とコスト削減

システム監視の統合
管理作業の自動化
トラブルシューティングの迅速化

- OpenTelemetryにより既存の監視ツールとシームレスに連携
- ミラーデータベース構築の簡素化により、IT部門の作業負荷を大幅に軽減
- 問題の根本原因を素早く特定し、ダウンタイムを最小化

堅牢なセキュリティと規制対応

きめ細かな監査制御
機密データの保護強化
規制コンプライアンスの容易な対応

- SQLアクセスの監視を強化し、内部統制要件に対応
- 自動暗号化により重要情報を守りながら、業務効率を維持
- 厳格なセキュリティ基準を満たし、監査への備えを強化

ヘルスケア組織向け特別機能

医療データの相互運用性向上
クラウドベースFHIRサービス
患者データの統合と活用

- FHIRサポートの拡充によりシステム間連携を強化
- 迅速な導入と柔軟なスケーラビリティを実現
- 包括的な患者ビューにより、ケアの質を向上

開発者エクスペリエンスの向上

SQL標準への完全準拠
エラー報告の強化
Copilot機能の拡充

- 既存スキルを活かした開発を可能に
- 問題の特定と修正の時間を短縮
- AIの力で開発生産性を大幅に向上



2025.1リリースについて

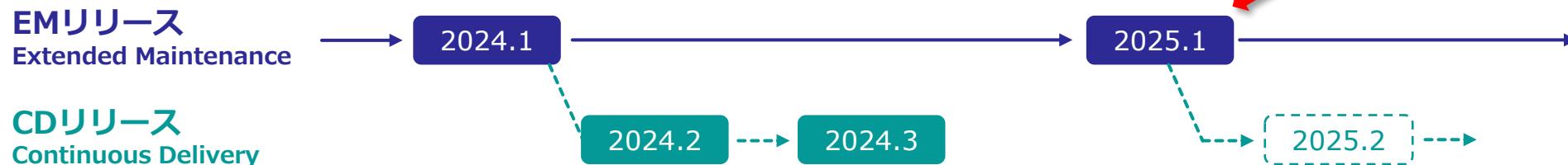
公式ドキュメント:2025.1リリースノート(英文)

https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GCRN_new20251

開発者コミュニティ

<https://jp.community.intersystems.com/post/intersystems-iris-intersystems-iris-health-%E3%81%8A%E3%82%88%E3%81%B3-healthshare-health-connect-20251-%E3%83%AA%E3%83%AA%E3%83%BC%E3%82%8B%E3%81%AE%E3%81%94%E6%A1%88%E5%86%85>

製品のリリースサイクル





2025.1実機デモ環境について

実機デモで利用する環境

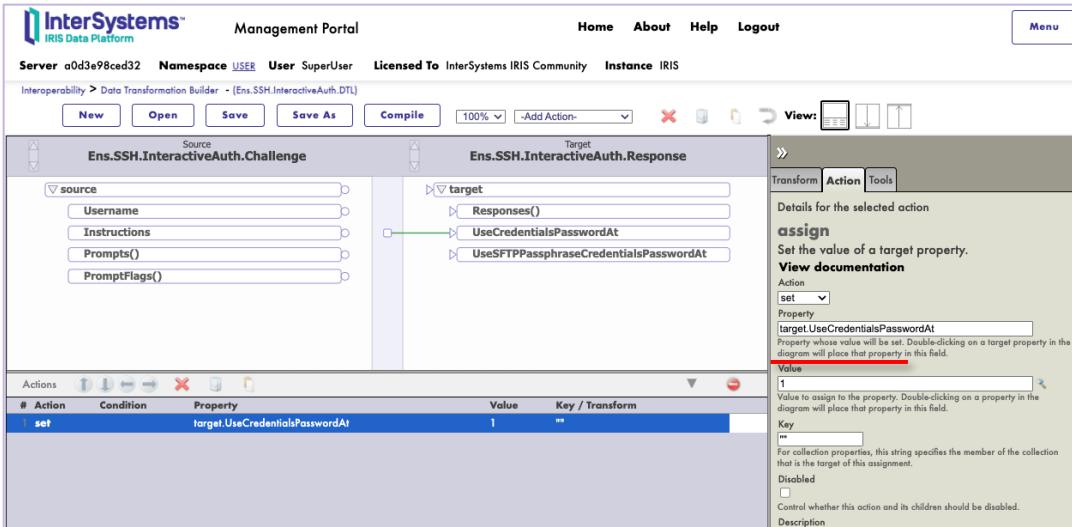
システム概要	
バージョン	IRIS for UNIX (Ubuntu Server LTS for ARM64 Containers) 2025.1 (Build 223U) Tue Mar 11 2025 18:22:49 EDT
構成	/usr/irissys/iris.cpf
データベースキャッシュ(MB)	4490
ルーチンキャッシュ (MB)	448
ジャーナルファイル	/usr/irissys/mgr/journal/20250423.001
スーパーサーバ・ポート	1972
ウェブサーバポート	52773
ライセンスサーバアドレス/ポート	/
ライセンス先	InterSystems IRIS Community
クラスタサポート	このシステムはクラスタの一部ではありません
ミラーリング	このシステムはミラーメンバではありません
システム開始日時	2025-04-20 14:05:20
暗号化キー識別子	利用可能ではありません。暗号化は有効になっていません。
NLSロケール	ENUW
このセッションの優先言語	日本語 ▾

Container Registryのlatest-previewからCommunity Editionを利用

開発者向け機能強化

DTLエディターにおけるPythonのサポート

2024.3



DTLでもPythonの便利ツールが使えるようになります

- ・ 数千種類あるPythonライブラリが使える
- ・ データ分析や機械学習のライブラリも活用可能
- ・ 複雑な処理も簡単に書ける

開発者の選択肢を増やし、仕事をもっと簡単に、早く、楽しくします。特にPythonの知識がある人にとっては、IRISでの開発がぐっと身近になります

Python Support in DTL Editor

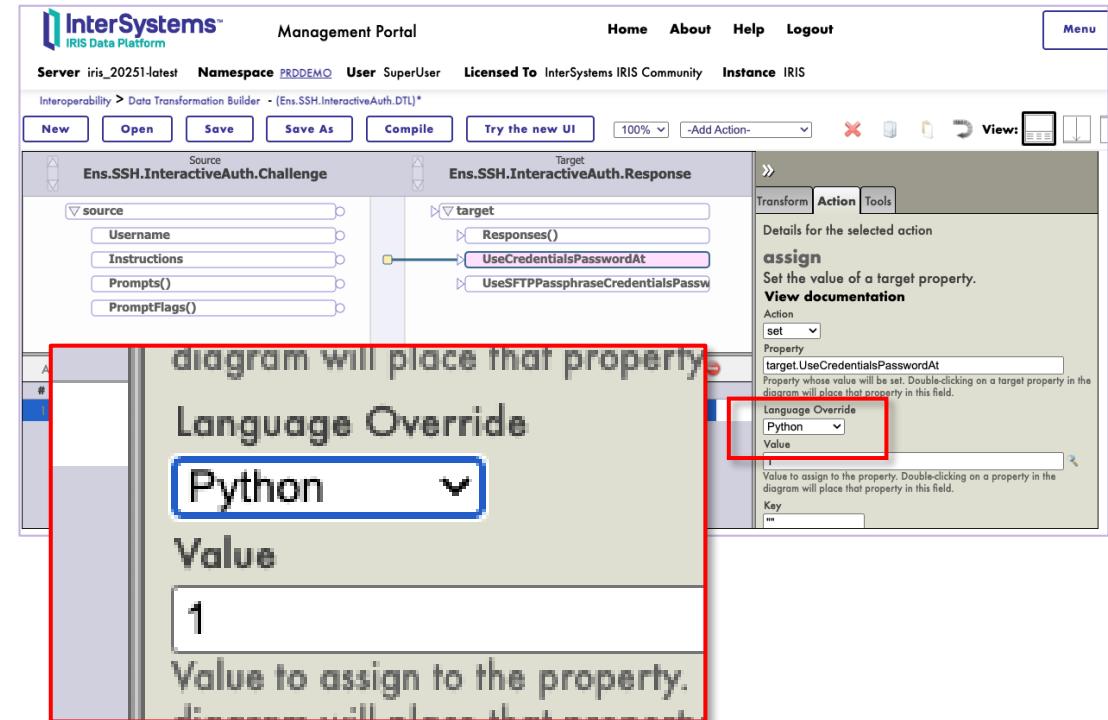
This release of InterSystems IRIS introduces the capability to leverage Python statements in your Data Transformation Language (DTL) classes. For each action, users can select the language (ObjectScript or Python) enabling developers who are skilled in Python to utilize larger parts of the platform with their existing skill set.



Interoperability



2025.1



DTLエディター新UIでのPython指定



DTL EDITOR

DTLDEMO.HospitalAtoUHR *

Hospital.A.PatientData Create new SET UHR.Patient

source Hospital.A.PatientData target UHR.Patient

```

graph LR
    subgraph source
        direction TB
        S1[patientNumber] --- T1[patientId]
        S2[personalInfo] --- T2[personalDetails]
        S3[lastName] --- T3[lastName]
        S4[firstName] --- T4[firstName]
        S5[dob] --- T5[dateOfBirth]
        S6[gender] --- T6[gender]
        S7[contactDetails] --- T7[contact]
        S8[insuranceInfo] --- T8[insurance]
        S9[provider] --- T9[provider]
        S10[policyNumber] --- T10[policyNumber]
        S11[validUntil] --- T11[validUntil]
        S12[medicalHistory] --- T12[encounters()]
        S13[visitDate] --- T13[encounterDate]
        S14[department] --- T14[clinician]
        S15[doctorName] --- T15[department]
        S16[symptoms] --- T16[symptoms]
        S17[diagnosis] --- T17[diagnosis]
        S18[medication] --- T18[medications()]
        S19[medication] --- T19[metadata]
        S20[lastUpdated] --- T20[sourceSystem]
    end
    subgraph target
        direction TB
        T1[patientId]
        T2[personalDetails]
        T3[lastName]
        T4[firstName]
        T5[dateOfBirth]
        T6[gender]
        T7[contact]
        T8[insurance]
        T9[provider]
        T10[policyNumber]
        T11[validUntil]
        T12[encounters()]
        T13[encounterDate]
        T14[clinician]
        T15[department]
        T16[symptoms]
        T17[diagnosis]
        T18[medications()]
        T19[metadata]
        T20[sourceSystem]
    end
    S1 --- T1
    S2 --- T2
    S3 --- T3
    S4 --- T4
    S5 --- T5
    S6 --- T6
    S7 --- T7
    S8 --- T8
    S9 --- T9
    S10 --- T10
    S11 --- T11
    S12 --- T12
    S13 --- T13
    S14 --- T14
    S15 --- T15
    S16 --- T16
    S17 --- T17
    S18 --- T18
    S19 --- T19
    S20 --- T20

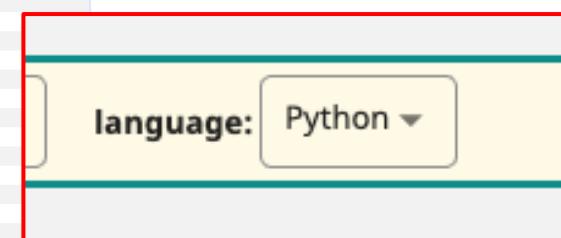
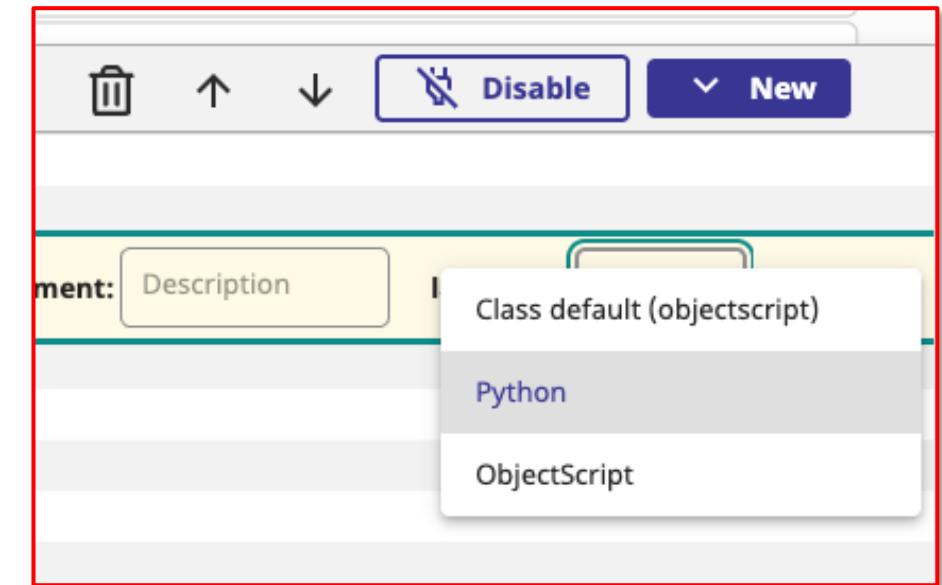
```

Actions (1 selected)

```

1 [SET] target.patientId = ##class(UHR.Util.DTLFunctions).FormatUP(source.patientNumber) 病院Aの患者番号を統一患者IDに変換
2 [SET] target.personalDetails.lastName = source.personalInfo.lastName
3 [SET] target.personalDetails.firstName source: source.personalInfo.firstName key: "" comment: Description language: Python
4 [SET] target.personalDetails.dateOfBirth = ##class(UHR.Util.DTLFunctions).FormatJDate(source.personalInfo.dob)
5 [SWITCH] 性別コードを標準形式に変換
6 [CASE] source.personalInfo.gender = "M"
7 [SET] target.personalDetails.gender = male
8 [CASE] source.personalInfo.gender = "F"
9 [SET] target.personalDetails.gender = female
10 [DEFAULT]
11 [SET] target.personalDetails.gender = unknown
12 [SET] target.personalDetails.contact.phoneNumber = source.personalInfo.contactDetails.phone
13 [SET] target.personalDetails.contact.email = source.personalInfo.contactDetails.email
14 [SET] target.personalDetails.contact.address = source.personalInfo.contactDetails.address
15 [SET] target.insurance.provider = source.insuranceInfo.provider
16 [SET] target.insurance.policyNumber = source.insuranceInfo.policyNumber
17 [SET] target.insurance.validUntil = ##class(UHR.Util.DTLFunctions).FormatJDate(source.insuranceInfo.validUntil)
18 [SET] target.metadata.lastUpdated = ##class(UHR.Util.DTLFunctions).FormatDateTime(source.lastUpdated)
19 [SET] target.metadata.originalId = source.patientNumber
20 [SET] target.metadata.sourceSystem = "HospitalA_EHR_v2.5"

```



AIおよびアナリティクスの機能強化

ベクトル検索における近似最近傍探索(ANN)インデックスサポート

- 人気の高いHierarchical Navigable Small World (HNSW)アルゴリズムを採用
- 数百万のベクトルに対しても1秒未満の類似性検索応答時間を実現
- 1億ベクトルの検索時、インデックスなしの「総当たり」検索と比較して**約2,500倍**の高速化を達成
- インデックス作成時にユーザーは使用する類似性関数 (VECTOR_DOT_PRODUCTまたはVECTOR_COSINE) を選択可能
- %EMBEDDINGインデックスと組み合わせて使用可能

対象：50万件のcc100-ja日本語データセットとベクトル可(Embed)したデータ

モデル：intfloat/multilingual-e5-large-instruct(1024次元でEmbed)

#	columns	vector
1	id	[REDACTED]
2	content	[REDACTED]
3	vector	[REDACTED]
4	vector	[REDACTED]
5	vector	[REDACTED]
6	vector	[REDACTED]
7	vector	[REDACTED]
8	vector	[REDACTED]
9	vector	[REDACTED]
10	vector	[REDACTED]
11	vector	[REDACTED]
12	vector	[REDACTED]
13	vector	[REDACTED]
14	vector	[REDACTED]
15	vector	[REDACTED]
16	vector	[REDACTED]
17	vector	[REDACTED]
18	vector	[REDACTED]
19	vector	[REDACTED]
20	vector	[REDACTED]
21	vector	[REDACTED]
22	vector	[REDACTED]
23	vector	[REDACTED]
24	vector	[REDACTED]
25	vector	[REDACTED]
26	vector	[REDACTED]
27	vector	[REDACTED]
28	vector	[REDACTED]
29	vector	[REDACTED]
30	vector	[REDACTED]
31	vector	[REDACTED]
32	vector	[REDACTED]
33	vector	[REDACTED]
34	vector	[REDACTED]
35	vector	[REDACTED]
36	vector	[REDACTED]
37	vector	[REDACTED]
38	vector	[REDACTED]
39	vector	[REDACTED]
40	vector	[REDACTED]
41	vector	[REDACTED]
42	vector	[REDACTED]
43	vector	[REDACTED]
44	vector	[REDACTED]
45	vector	[REDACTED]
46	vector	[REDACTED]
47	vector	[REDACTED]
48	vector	[REDACTED]
49	vector	[REDACTED]
50	vector	[REDACTED]
51	vector	[REDACTED]
52	vector	[REDACTED]
53	vector	[REDACTED]
54	vector	[REDACTED]
55	vector	[REDACTED]
56	vector	[REDACTED]
57	vector	[REDACTED]
58	vector	[REDACTED]
59	vector	[REDACTED]
60	vector	[REDACTED]
61	vector	[REDACTED]
62	vector	[REDACTED]
63	vector	[REDACTED]
64	vector	[REDACTED]
65	vector	[REDACTED]
66	vector	[REDACTED]
67	vector	[REDACTED]
68	vector	[REDACTED]
69	vector	[REDACTED]
70	vector	[REDACTED]
71	vector	[REDACTED]
72	vector	[REDACTED]
73	vector	[REDACTED]
74	vector	[REDACTED]
75	vector	[REDACTED]
76	vector	[REDACTED]
77	vector	[REDACTED]
78	vector	[REDACTED]
79	vector	[REDACTED]
80	vector	[REDACTED]
81	vector	[REDACTED]
82	vector	[REDACTED]
83	vector	[REDACTED]
84	vector	[REDACTED]
85	vector	[REDACTED]
86	vector	[REDACTED]
87	vector	[REDACTED]
88	vector	[REDACTED]
89	vector	[REDACTED]
90	vector	[REDACTED]
91	vector	[REDACTED]
92	vector	[REDACTED]
93	vector	[REDACTED]
94	vector	[REDACTED]
95	vector	[REDACTED]
96	vector	[REDACTED]
97	vector	[REDACTED]
98	vector	[REDACTED]
99	vector	[REDACTED]
100	vector	[REDACTED]
101	vector	[REDACTED]
102	vector	[REDACTED]
103	vector	[REDACTED]
104	vector	[REDACTED]
105	vector	[REDACTED]
106	vector	[REDACTED]
107	vector	[REDACTED]
108	vector	[REDACTED]
109	vector	[REDACTED]
110	vector	[REDACTED]
111	vector	[REDACTED]
112	vector	[REDACTED]
113	vector	[REDACTED]
114	vector	[REDACTED]
115	vector	[REDACTED]
116	vector	[REDACTED]
117	vector	[REDACTED]
118	vector	[REDACTED]
119	vector	[REDACTED]
120	vector	[REDACTED]
121	vector	[REDACTED]
122	vector	[REDACTED]
123	vector	[REDACTED]
124	vector	[REDACTED]
125	vector	[REDACTED]
126	vector	[REDACTED]
127	vector	[REDACTED]
128	vector	[REDACTED]
129	vector	[REDACTED]
130	vector	[REDACTED]
131	vector	[REDACTED]
132	vector	[REDACTED]
133	vector	[REDACTED]
134	vector	[REDACTED]
135	vector	[REDACTED]
136	vector	[REDACTED]
137	vector	[REDACTED]
138	vector	[REDACTED]
139	vector	[REDACTED]
140	vector	[REDACTED]
141	vector	[REDACTED]
142	vector	[REDACTED]
143	vector	[REDACTED]
144	vector	[REDACTED]
145	vector	[REDACTED]
146	vector	[REDACTED]
147	vector	[REDACTED]
148	vector	[REDACTED]
149	vector	[REDACTED]
150	vector	[REDACTED]
151	vector	[REDACTED]
152	vector	[REDACTED]
153	vector	[REDACTED]
154	vector	[REDACTED]
155	vector	[REDACTED]
156	vector	[REDACTED]
157	vector	[REDACTED]
158	vector	[REDACTED]
159	vector	[REDACTED]
160	vector	[REDACTED]
161	vector	[REDACTED]
162	vector	[REDACTED]
163	vector	[REDACTED]
164	vector	[REDACTED]
165	vector	[REDACTED]
166	vector	[REDACTED]
167	vector	[REDACTED]
168	vector	[REDACTED]
169	vector	[REDACTED]
170	vector	[REDACTED]
171	vector	[REDACTED]
172	vector	[REDACTED]
173	vector	[REDACTED]
174	vector	[REDACTED]
175	vector	[REDACTED]
176	vector	[REDACTED]
177	vector	[REDACTED]
178	vector	[REDACTED]
179	vector	[REDACTED]
180	vector	[REDACTED]
181	vector	[REDACTED]
182	vector	[REDACTED]
183	vector	[REDACTED]
184	vector	[REDACTED]
185	vector	[REDACTED]
186	vector	[REDACTED]
187	vector	[REDACTED]
188	vector	[REDACTED]
189	vector	[REDACTED]
190	vector	[REDACTED]
191	vector	[REDACTED]
192	vector	[REDACTED]
193	vector	[REDACTED]
194	vector	[REDACTED]
195	vector	[REDACTED]
196	vector	[REDACTED]
197	vector	[REDACTED]
198	vector	[REDACTED]
199	vector	[REDACTED]
200	vector	[REDACTED]
201	vector	[REDACTED]
202	vector	[REDACTED]
203	vector	[REDACTED]
204	vector	[REDACTED]
205	vector	[REDACTED]
206	vector	[REDACTED]
207	vector	[REDACTED]
208	vector	[REDACTED]
209	vector	[REDACTED]
210	vector	[REDACTED]
211	vector	[REDACTED]
212	vector	[REDACTED]
213	vector	[REDACTED]
214	vector	[REDACTED]
215	vector	[REDACTED]
216	vector	[REDACTED]
217	vector	[REDACTED]
218	vector	[REDACTED]
219	vector	[REDACTED]
220	vector	[REDACTED]
221	vector	[REDACTED]
222	vector	[REDACTED]
223	vector	[REDACTED]
224	vector	[REDACTED]
225	vector	[REDACTED]
226	vector	[REDACTED]
227	vector	[REDACTED]
228	vector	[REDACTED]
229	vector	[REDACTED]
230	vector	[REDACTED]
231	vector	[REDACTED]
232	vector	[REDACTED]
233	vector	[REDACTED]
234	vector	[REDACTED]
235	vector	[REDACTED]
236	vector	[REDACTED]
237	vector	[REDACTED]
238	vector	[REDACTED]
239	vector	[REDACTED]
240	vector	[REDACTED]
241	vector	[REDACTED]
242	vector	[REDACTED]
243	vector	[REDACTED]
244	vector	[REDACTED]
245	vector	[REDACTED]
246	vector	[REDACTED]
247	vector	[REDACTED]
248	vector	[REDACTED]
249	vector	[REDACTED]
250	vector	[REDACTED]
251	vector	[REDACTED]
252	vector	[REDACTED]
253	vector	[REDACTED]
254	vector	[REDACTED]
255	vector	[REDACTED]
256	vector	[REDACTED]
257	vector	[REDACTED]
258	vector	[REDACTED]
259	vector	[REDACTED]
260	vector	[REDACTED]
261	vector	[REDACTED]
262	vector	[REDACTED]
263	vector	[REDACTED]
264	vector	[REDACTED]
265	vector	[REDACTED]
266	vector	[REDACTED]
267	vector	[REDACTED]
268	vector	[REDACTED]
269	vector	[REDACTED]
270	vector	[REDACTED]
271	vector	[REDACTED]
272	vector	[REDACTED]
273	vector	[REDACTED]
274	vector	[REDACTED]
275	vector	[REDACTED]
276	vector	[REDACTED]
277	vector	[REDACTED]
278	vector	[REDACTED]
279	vector	[REDACTED]
280	vector	[REDACTED]
281	vector	[REDACTED]
282	vector	[REDACTED]
283	vector	[REDACTED]
284	vector	[REDACTED]
285	vector	[REDACTED]
286	vector	[REDACTED]
287	vector	[REDACTED]
288	vector	[REDACTED]
289	vector	[REDACTED]
290	vector	[REDACTED]
291	vector	[REDACTED]
292	vector	[REDACTED]
293	vector	[REDACTED]
294	vector	[REDACTED]
295	vector	[REDACTED]
296	vector	[REDACTED]
297	vector	[REDACTED]
298	vector	[REDACTED]
299	vector	[REDACTED]
300	vector	[REDACTED]
301	vector	[REDACTED]
302	vector	[REDACTED]
303	vector	[REDACTED]
304	vector	[REDACTED]
305	vector	[REDACTED]
306	vector	[REDACTED]
307	vector	[REDACTED]
308	vector	[REDACTED]
309	vector	[REDACTED]
310	vector	[REDACTED]
311	vector	[REDACTED]
312	vector	[REDACTED]
313	vector	[REDACTED]
314	vector	[REDACTED]
315	vector	[REDACTED]
316	vector	[REDACTED]
317	vector	[REDACTED]
318	vector	[REDACTED]
319	vector	[REDACTED]
320	vector	[REDACTED]
321	vector	[REDACTED]
322	vector	[REDACTED]
323	vector	[REDACTED]
324	vector	[REDACTED]
325	vector	[REDACTED]
326	vector	[REDACTED]
327	vector	[REDACTED]
328	vector	[REDACTED]
329	vector	[REDACTED]
330	vector	[REDACTED]
331	vector	[REDACTED]
332	vector	[REDACTED]
333	vector	[REDACTED]
334	vector	[REDACTED]
335	vector	[REDACTED]
336	vector	[REDACTED]
337	vector	[REDACTED]
338	vector	[REDACTED]
339	vector	[REDACTED]
340	vector	[REDACTED]
341	vector	[REDACTED]
342	vector	[REDACTED]
343	vector	[REDACTED]
344	vector	[REDACTED]
345	vector	[REDACTED]
346	vector	[REDACTED]
347	vector	[REDACTED]
348	vector	[REDACTED]
349	vector	[REDACTED]
350	vector	[REDACTED]
351	vector	[REDACTED]
352	vector	[REDACTED]
353	vector	[REDACTED]
354	vector	[REDACTED]
355	vector	[REDACTED]
356	vector	[REDACTED]
357	vector	[REDACTED]
358	vector	[REDACTED]
359	vector	[REDACTED]
360	vector	[REDACTED]
361	vector	[REDACTED]
362	vector	[REDACTED]
363	vector	[REDACTED]
364	vector	[REDACTED]
365	vector	[REDACTED]
366	vector	[REDACTED]
367	vector	[REDACTED]
368	vector	[REDACTED]
369	vector	[REDACTED]
370	vector	[REDACTED]
371	vector	[REDACTED]
372	vector	[REDACTED]
373	vector	[REDACTED]
374	vector	[REDACTED]
375	vector	[REDACTED]
376	vector	[REDACTED]
377	vector	[REDACTED]
378	vector	[REDACTED]
379	vector	[REDACTED]
380	vector	[REDACTED]
381	vector	[REDACTED]
382	vector	[REDACTED]
383	vector	[REDACTED]
384	vector	[REDACTED]
385	vector	[REDACTED]
386	vector	[REDACTED]
387	vector	[REDACTED]
388	vector	[REDACTED]
389	vector	[



近似最近傍探索(ANN)インデックスによる結果の違い

検索方法：10万件のcc100-ja日本語データセットにて類似意味文章のCOSINE検索

インデックス無し

検索時間：**0.05996秒**

8. Similarity: 0.889039 (ID: 23579)
その傍らでは犠牲者の冥福を祈るように綺麗な虹が現れていたことが対照的であった。

9. Similarity: 0.887119 (ID: 16626)
気持ちがいっぱいいっぱいだったのもかけているかもしれません、いっぱいいっぱいの緑色の光、彼女の未来を照らす光、いいじゃない。

10. Similarity: 0.886722 (ID: 56030)
痛い子さんは、自分で気付かないうちに辛いこと、悲しいことを全部自分の中にため込んでしまったんですね。

17. Similarity: 0.883067 (ID: 40489)
女子高生は身の潔白を主張するために涙を振り絞りながら制服を脱いで主人公に近づいていきます。

18. Similarity: 0.883029 (ID: 20484)
太陽は昼明るい場所に出ているため、今後は彼女の心がスッキリと晴れていこうとの様子を表すものであると思いました。

19. Similarity: 0.882882 (ID: 62071)
これ以上ない悲しみの日々の中で、学びながら成長して、強くなつたのですね。

ANNインデックス有り

検索時間：**0.01805秒**

8. Similarity: 0.889039 (ID: 23579)
その傍らでは犠牲者の冥福を祈るように綺麗な虹が現れていたことが対照的であった。

9. Similarity: 0.886722 (ID: 56030)
痛い子さんは、自分で気付かないうちに辛いこと、悲しいことを全部自分の中にため込んでしまったんですね。

16. Similarity: 0.883067 (ID: 40489)
女子高生は身の潔白を主張するために涙を振り絞りながら制服を脱いで主人公に近づいていきます。

17. Similarity: 0.882882 (ID: 62071)
これ以上ない悲しみの日々の中で、学びながら成長して、強くなつたのですね。

注意！：検索漏れが発生する可能性がある

AIおよびアナリティクスの機能強化

IRIS BIのビルドと同期処理を改善

自動依存関係分析	キューブを更新する際、関連する他のキューブも自動的に特定し、適切な順序で更新するようになりました
一貫性の確保	複数の関連キューブ間でデータの整合性が自動的に維持されるようになりました
新関数の追加	個別のキューブ更新のための新しい関数（%BuildOneCubeと%SynchronizeOneCube）が導入されました
詳細なログ記録	全てのキューブ更新過程が記録されるようになりました、問題発生時の原因究明が容易になりました

効果的な利用シナリオ



医療業界



複数の医療機関からのデータを統合し分析する際、データソースごとのキューブが適切に更新され、研究結果の信頼性が高まります。例えば、新しい臨床試験データが追加された場合、関連する全ての分析キューブが正確に更新されます。

Updates to Business Intelligence Cube Building and Synchronization

This release includes changes to the InterSystems IRIS Business Intelligence cube building process to provide safeguards for updating related cube dependency chains and other enhancements that address several customer-reported issues in this area. To ensure that analysis of cube builds is always taken into consideration and that data remains consistent, the base functions for cube building and synchronization (%BuildCube and %SynchronizeCube) now perform cube dependency analysis automatically and execute a full model group update. This new procedure includes:



物流業界



販売データキューブ、在庫レベルキューブ、季節トレンドキューブなどが連動して更新されることで、より精度の高い需要予測と適切な在庫計画が可能になります。



SQLおよびデータ管理の機能強化

サーバ側ページ処理に適した新たなSQL文法

InterSystems IRISのSQLエンジンが大きく進化し、クエリ結果のページネーション処理において新たな構文オプションが追加されました。

2024.3

従来、InterSystems IRIS SQLでは結果セットを制限するためTOPキーワードのみをサポートしていました：

```
SQL> SELECT TOP 10 Name, Age FROM Patients
ORDER BY Age DESC
```

この場合、ECサイトなどで複数ページを必要とする商品リストやコメントなどを表示する場合、大量のデータをSELECT ... TOP 10した後に不要な部分を表示させない、などの処理が必要で、不必要にシステムの検索負荷を高めてしまうことが問題でした。

New syntax for server-side pagination

InterSystems IRIS SQL has long supported the TOP keyword to limit query results only to the first n records of a sorted result set. With this release, InterSystems adds support for two alternative syntax flavors: LIMIT ... OFFSET ..., which is commonly used in other database platforms, and OFFSET ... FETCH ..., which is the official ANSI standard. By offering all three methods of server-side pagination, InterSystems has made it easier for new customers and users migrating to InterSystems IRIS to write efficient queries.

2025.1

2025.1では、以下の2つの新しい構文がサポートされました：

- LIMIT ... OFFSET ... 構文
多くのデータベース（MySQL、PostgreSQL、SQLiteなど）で広く採用されている構文です：

```
SQL> SELECT Name, Age FROM Patients ORDER BY Age DESC
LIMIT 10 OFFSET 20
```

- OFFSET ... FETCH ... 構文
ANSI SQLの標準構文です：

```
SQL> SELECT Name, Age FROM Patients ORDER BY Age DESC
OFFSET 20 ROWS FETCH NEXT 10 ROWS ONLY
```

SQLおよびデータ管理の機能強化

DDL文の一括インポート機能

これまで、大量データのロードにはLOAD DATAコマンドが利用できていましたが、テーブル、ビューやインデックスの作成などのDDLは個別に実施する必要がありました。

2025.1では、大量のテーブルやビュー、インデックスなどを作成したり、ユーザの作成や権限の付与など、システムの導入時に面倒だった個別のDDLの実行を自動化するLOAD SQLコマンドを搭載しました。

テーブルを作成するLOAD SQLコマンド例：

```
LOAD SQL FROM FILE '/opt/iris/loadsqI/sql/schema/tables/tables.sql'
DIALECT 'IRIS'
LOG TO FILE '/opt/iris/loadsqI/logs/tables_log.txt'
IGNORE ERRORS
```

インデックスを作成するLOAD SQLコマンド例：

```
LOAD SQL FROM FILE '/opt/iris/loadsqI/sql/schema/indexes/indexes.sql'
DIALECT 'IRIS'
DELIMITER ;
VERBOSE LOG TO FILE '/opt/iris/loadsqI/logs/indexes_log.txt'
```

Better bulk Import of DDL statements

Based on customer feedback on the LOAD DATA command, InterSystems IRIS SQL now supports a similar LOAD SQL command to load and execute any number of SQL DDL statements from a file or directory. This new command offers a simple entry point for developers accessing the system over SQL and writes detailed output, either to the SQL DiagnosticsOpens in a new tab table or another log file. The pre-existing ObjectScript entry pointsOpens in a new tab have also been enhanced to support the diagnostic output.



注意点：

- LOAD SQLコマンドをLOAD SQLコマンドから実行することはできません。
- ディレクトリに含まれる全てのSQLファイルをインポートし実行することができますが、依存関係があるものについては順番を考慮し実行する必要があります。
- ビットスライスインデックスの作成はできません。

【 次ページに具体的なDDL SQLファイルが記載されています 】

LOAD SQLコマンド：DDL SQLファイル内容

tables.sql

```
CREATE TABLE Customer (
    CustomerID INT PRIMARY KEY,
    Name VARCHAR(100),
    Email VARCHAR(100),
    Phone VARCHAR(20),
    Address VARCHAR(200),
    RegistrationDate DATE
)
GO
```

```
CREATE TABLE Product (
    ProductID INT PRIMARY KEY,
    Name VARCHAR(100),
    Description TEXT,
    Price DECIMAL(10,2),
    StockQuantity INT,
    Category VARCHAR(50)
)
GO
```

```
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATETIME,
    TotalAmount DECIMAL(10,2),
    Status VARCHAR(20),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
)
GO
```

```
CREATE TABLE OrderDetail (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT,
    UnitPrice DECIMAL(10,2),
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
)
GO
```

indexes.sql

```
CREATE INDEX idx_customer_name ON Customer (Name);
CREATE INDEX idx_customer_email ON Customer (Email);
CREATE INDEX idx_product_name ON Product (Name);
CREATE INDEX idx_product_category ON Product (Category);
CREATE INDEX idx_order_customer ON Orders (CustomerID);
CREATE INDEX idx_order_date ON Orders (OrderDate);
CREATE INDEX idx_order_status ON Orders (Status);
CREATE INDEX idx_orderdetail_order ON OrderDetail (OrderID);
CREATE INDEX idx_orderdetail_product ON OrderDetail (ProductID);
```

sample_data.sql

```
INSERT INTO Customer(CustomerID, Name, Email, Phone, Address, RegistrationDate)
VALUES (1, '山田太郎', 'yamada@example.com', '03-1234-5678', '東京都新宿区新宿1-1-1', TO_DATE('2024-01-15', 'YYYY-MM-DD'));

INSERT INTO Customer (CustomerID, Name, Email, Phone, Address, RegistrationDate)
VALUES (2, '佐藤花子', 'sato@example.com', '03-2345-6789', '東京都渋谷区渋谷2-2-2', TO_DATE('2024-02-20', 'YYYY-MM-DD'));

INSERT INTO Customer (CustomerID, Name, Email, Phone, Address, RegistrationDate)
VALUES (3, '鈴木一郎', 'suzuki@example.com', '03-3456-7890', '東京都品川区品川3-3-3', TO_DATE('2024-03-10', 'YYYY-MM-DD'));

-- Product データ
INSERT INTO Product (ProductID, Name, Description, Price, StockQuantity, Category)
VALUES (101, 'ノートパソコン', '高性能ノートパソコン', 120000.00, 50, '電子機器');

INSERT INTO Product (ProductID, Name, Description, Price, StockQuantity, Category)
VALUES (102, 'スマートフォン', '最新モデルスマートフォン', 80000.00, 100, '電子機器');

INSERT INTO Product (ProductID, Name, Description, Price, StockQuantity, Category)
VALUES (103, 'ワイヤレスイヤホン', 'ノイズキャンセリング機能付き', 15000.00, 200, 'オーディオ');

INSERT INTO Product (ProductID, Name, Description, Price, StockQuantity, Category)
VALUES (104, 'スマートウォッチ', '健康管理機能付き', 25000.00, 75, 'ウェアラブル');
```

※DML文の実行も可能です。





SQLおよびデータ管理の機能強化

追加されたSQLファンクション

APPROX_COUNT_DISTINCT()

- 処理速度の大幅向上：HyperLogLogアルゴリズムを実装しており、従来の COUNT(DISTINCT) と比較して桁違いに高速に処理できます。
- メモリ効率の改善：大規模データセットでも一定のメモリ使用量で動作するため、リソース使用量を抑えられます。
- スケーラビリティの向上：データ量が増加しても処理時間がほぼ一定で、膨大なデータに対しても効率的です。
- リアルタイム分析の実現：高速な近似計算により、対話的なデータ分析やダッシュボードでの使用に適しています。
- 誤差許容の柔軟性：完全な正確性よりも速度を優先するケースで有用です（誤差率は通常2~5%程度）。

対象：1,000万件のダミーデータ

#	ID	Name	Address	Age	Occupation
1	1	池田 治	神奈川県横浜市港北区本町新田23丁目17番11号 木工クリスト238	87	エステティシャン
2	2	渡辺 雅	滋賀県守口市守口本町42丁目18番9号 コート無葉屋043	34	靴職人
3	3	山崎 茂子	富山県高岡市鶴見町5丁目18番10号 上吉羽パーク860	81	音楽家
4	4	山口 あすか	埼玉県熊谷市中央通り3丁目2番9号 リミチャーム376	67	ウェディングプランナー
5	5	山崎 太一	鳥取県米子市中央通り5丁目20番9号 独絶特53	44	アニメーター
6	6	小林 丈人	広島県呉市中央木立2丁目2番18号 バー秋葉原022	52	介護ヘルパー
7	7	佐々木 亮介	長崎県佐世保市東町島原24丁目6番9号 コート848	24	占い師
8	8	橋本 寿美子	山口県下関市多摩久丸の内Jアーバー22丁目15番5号 平野町アーバン059	59	奉司職人
9	9	佐々木 知実	宮城県仙台市中崎9丁目25番4号 銀治の沢バス876	77	行政書士
10	10	松田 知実	東京都あきる野市高輪32丁目22番9号 コーポ御嶽和94	57	歯科医師
11	11	伊藤 知実	千葉県浦安市上野木12丁目20番5号 ハイツ宮崎028	74	イラストレーター
12	12	佐藤 あいか	沖縄県那覇市栄町4丁目1番17号	33	運転士
13	13	藤井 和樹	岩手県大崎市虎ノ門4丁目1番18号 天神島シティ852	82	お笑い芸人
14	14	田中 知美子	愛媛県東温市久米町浦23丁目4番4号 前坪コート412	19	栄養士
15	15	松本 知衣	三重県桑名市手向町125番12号 南郷園シティ696	20	医師
16	16	清水 加弘	山形県最上郡山市萬草橋12丁目24番2号	79	医療事務員
17	17	吉田 知緒	宮城県千代田区大町24丁目18番3号 ニコボ平上000	55	配管工
18	18	山田 丈人	大分県大分市白子町羽折町35丁目3番1号 コーポ平河町411	78	看護師
19	19	小川 花子	岩手県奥州市水沢31丁目1番20号 ハイツ中宮町920	34	イラストレーター
20	20	阿部 知	広島県東陽町光40丁目4番4号	89	小説家
21	21	田中 治	山形県南陽市舟町8丁目24番9号	34	行政書士
22	22	加藤 健一	富山県富山市宇波坂北上野22丁目25番9号 来洗草シヤム324	51	エンジニア演奏家
23	23	小林 舞	高知県香川郡多古町元洗2丁目19番17号 バス洗車187	32	アナウンサー
24	24	鈴木 舞	奈良県生駒郡鈴鹿町前六丁目17丁目26番4号 パーク田辺562	81	映画監督
25	25	高橋 朝樹	栃木県宇都宮市船野8丁目12番12号	55	配管工
26	26	小林 舞	栃木県宇都宮市船野2丁目2番9号 コート前弥6135	22	グラフィックデザイナー
27	27	佐々木 太一	福島県喜多方市喜多方31丁目18番11号 バークアーバン395	26	アナウンサー
28	28	田中 みみ子	千葉県柏市柏丸の内28丁目6番11号 バークアーバン030	64	バスガイド
29	29	山田 学	富山県富山市南中町16丁目22番9号 イバタ島947	34	大学教授
30	30	山口 淳	滋賀県守口市禁煙本禁煙2丁目1番7号 シルム津久戸町726	50	作家
31	31	渡辺 充	岐阜県岐阜市鶴見町台場4丁目1番12号 パーク池の端833	27	医師
32	32	山崎 丈樹	鳥取県米子市宮本幸手2丁目2番15号	81	バスガイド
33	33	山崎 由子	鹿児島県鹿児島市大町7丁目27番5号 ハーバン西川186	66	絵本作家
34	34	山口 丈人	大分県大分市中東上野10丁目26番3号	85	介護ヘルパー
35	35	山本 也也	香川県高松市東上野22丁目22番11号	68	Youtuber
36	36	伊藤 加奈	群馬県東村山市東大和町21丁目5番8号 コート小糸川754	32	航海士

COUNT(DISTINCT (ITEM))

実行 ブラン表示 履歴を表示 クエリビルダ 表示モード ▾ 最大 1000 その他オプション

```
SELECT COUNT(DISTINCT ID) FROM "User".Person
```

Aggregate_1
10000000

IDの種類:1000万件

実行 ブラン表示 履歴を表示 クエリビルダ 表示モード ▾ 最大 1000 その他オプション

```
SELECT COUNT(DISTINCT Occupation) FROM "User".Person
```

Aggregate_1
56

職種:56件

Additional SQL functions

This release introduces two new convenience functions to the InterSystems SQL dialect:
`APPROX_COUNT_DISTINCT()` is a new aggregate function that implements the HyperLogLog algorithm, which offers users an estimate of the number of distinct values for a particular column that is orders of magnitude faster than the exact result obtained from a `COUNT(DISTINCT)` command.

APPROX_COUNT_DISTINCT(ITEM)

実行 ブラン表示 履歴を表示 クエリビルダ 表示モード ▾ 最大 1000 その他オプション

```
SELECT APPROX_COUNT_DISTINCT(ID) FROM "User".Person
```

Expression_1
1013197

IDの種類:約1013万件 誤差率:1.3%

実行 ブラン表示 履歴を表示 クエリビルダ 表示モード ▾ 最大 1000 その他オプション

```
SELECT APPROX_COUNT_DISTINCT(Occupation) FROM "User".Person
```

Expression_1
56

職種:訳56件

誤差率:0%



SQLおよびデータ管理の機能強化

追加されたSQLファンクション

HASHBYTES()

- セキュリティ強化：SHA256などの強力なハッシュアルゴリズムをSQL内から直接利用できるため、データのセキュリティが向上します。
- SQLからの直接アクセス：外部関数やプログラミング言語を使わずに、SQL内でハッシュ機能を利用できます。
- データ整合性の検証：大量のデータの整合性を効率的に検証できます。
- 暗号化処理の簡素化：パスワードハッシュやデータ保護において、実装が容易になります。
- データの匿名化：個人情報などのセンシティブなデータを匿名化する際に有用です。
- 重複検出の効率化：大きなデータセットでの重複検出において、ハッシュ値の比較が効率的です。

例えばこんな場合に・・・

- パスワードハッシュの保存
- データ整合性の検証
- センシティブデータの匿名化
- データマッチングと重複検出

実行
プラン表示
履歴を表示
クエリビルダ
表示モード
最大 1000
その他オプション

```
SELECT
    'テストデータ' AS ハッシュ対象,
    HASHBYTES('SHA256', 'テストデータ') AS sha256_hash,
    HASHBYTES('SHA512', 'テストデータ') AS sha512_hash;
```

Row count: 1 Performance: 0.0029 seconds 322 global references 3066 commands executed 0 disk read latency (ms) Query: %sqlcq.USER
14:14:42.399

ハッシュ対象	sha256_hash	sha512_hash
テストデータ	Ý ° cí:Uk ÷D%3 ck. OsÙùÖÙ Üř >äE¶ Mé©gi ÄÈä=x½X & ,äñ ö€öà YNÿ €‡:£ YÍ D +ê, i7j H5. ÍK]z M	



性能/拡張性/セキュリティの強化

ジャーナルコードのサイズ削減

1. \$INCREMENTの拡張性向上 :

1. \$INCREMENT関数（カウンター操作などに使用される関数）のスケーラビリティが向上しました
2. これにより、高負荷環境でのパフォーマンスが改善されます

2. 新しいジャーナルファイル形式の導入 :

1. より小さくコンパクトなジャーナルファイル形式が採用されました
2. ジャーナルコードのサイズが小さくなつたため、I/O帯域幅の使用が減少しパフォーマンスが改善されます
3. \$INCREMENTの記録方法も変更されています

注意点 :

- 2025.1で作成されたジャーナルファイルは、以前のバージョンのIRISでは開けません
- ミラーリング構成の場合は、どのミラーメンバーがアップグレードされたかを把握しておく必要があります
- ジャーナルファイルを読み取る独自ロジックを実装しているアプリケーションは、新しい\$INCREMENT処理の影響を確認する必要があります（ただし、多くの場合は影響ありません）

Smaller journal records

The 2025.1 release improves the scalability of \$INCREMENT and introduces a new, more compact journal file format. The new format uses less I/O bandwidth (due to the smaller journal record size) on average and includes changes to the way that \$INCREMENT is recorded in the journal. Journal files created on a 2025.1 instance cannot be opened on an instance running a prior release, so be sure to keep track of which mirror members are upgraded. Applications that have implemented custom logic to read and interpret journal files should be reviewed to determine whether this logic is impacted by the new \$INCREMENT handling discussed in Special Considerations When Upgrading; most such custom logic is not impacted.

補足 : \$incrementの動作

```

USER>write $increment(a)
1
USER>write $increment(a)
2
USER>write $increment(a)
3
USER>write $increment(a,5)
8
USER>write a
8
USER>write $increment(b)
1
USER>write $increment(c)
1
USER>write $increment(b)
2
USER>w a
8
USER>w b
2
USER>w c
1

```



性能/拡張性/セキュリティの強化

データベースコンパクション(圧縮)処理の高速化

データベースのコンパクション(圧縮)とデフラグメンテーション(断片化解消)の効率を高めるための多数の機能強化が含まれました。特に、**大量の大きな文字列を持つ大規模データベース**では、コンパクションプロセスが大幅に高速化され、アプリケーションが実運用環境下においても、パフォーマンス低下を最小限に抑えられるよう改善されました。

なお、コンパクションに要する時間は環境によって大きく異なり、データベースのサイズ(容量)、データの質、ハードウェア性能、現状のシステム負荷などが大きく影響を与えます。

一般的な目安としては以下となりますが、保証するものではない点にご注意ください。

小規模(~10GB) : 数秒~数十分

中規模(~100GB) : 数十分~数時間

大規模(100GB以上) : 数時間~数日

注意点 :

- データベースのコンパクションは実運用環境下でも実行できますが、システムの負荷が非常に高い状態が長時間続く場合、コンパクションにも長時間を要するか、完了しないことがあります。
- また、コンパクションの実行中に対象のデータベースのサイズが一時的に増加することがあります。データベースが設定された最大サイズに達した場合は、操作はキャンセルされます。

参考「データベースファイルのサイズを圧縮する方法」
<https://jp.community.intersystems.com/node/502001>

Faster database compaction

This release includes a number of enhancements to increase the efficiency of database compaction. It opens in a new tab and defragmentation. Especially for large databases with lots of big strings, the compaction process now runs significantly faster and is better aligned with any concurrent block allocation for application workloads.

%SYS>do ^DATABASE

- 1) Create a database
- 2) Edit a database
- 3) List databases
- 4) Delete a database
- 5) Mount a database
- 6) Dismount a database
- 7) Compact globals in a database ←
- 8) Show free space for a database
- 9) Show details for a database
- 10) Recreate a database
- 11) Manage database encryption
- 12) Return unused space for a database
- 13) Compact free space in a database
- 14) Defragment a database
- 15) Show background database tasks
- 16) Manage multi-volume database properties

Option? 7

Database directories to compact? /usr/irissys/mgr/user

Databases Selected

 /usr/irissys/mgr/user/

All Globals? Yes => Yes

47 items selected from

47 available globals

How full do you want the database blocks? 90 =>

Display compacted globals? No =>

Device:

Right margin: 80 =>

Confirm compaction of databases? Yes =>

Database Completed	Database Compaction
-----	Apr 10 2025 2:20 PM
-----	MB Processed MB Compressed To

/usr/irissys/mgr/user/	.287	.287	2:20 PM
------------------------	------	------	---------



性能/拡張性/セキュリティの強化

ミラーデータベースの自動配布機能

ミラーデータベースのダウンロード機能が実装され、データベースのミラーリング設定が簡単になりました。

従来の方法:

- データベースファイルを手動で他のミラーメンバーにコピーする必要があった
- APIコールを調整してデータベースを登録する手順が必要だった

新機能の特徴:

- 新しいデータベースが全ミラーメンバーに自動的に作成される
- グローバルデータが自動的にダウンロードされる
- ファイルコピーの手順が不要になった

メリット:

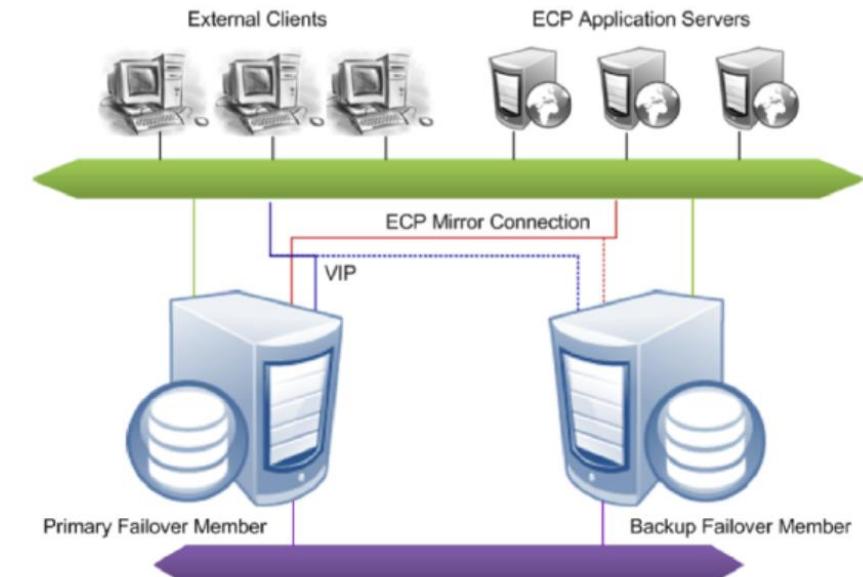
- 複雑なミラー環境のセットアップが大幅に簡略化

注意点 :

限られた時間でミラー構成のセットアップを実施する必要がある場合は、従来の手動コピー方式の方が、短時間でセットアップを完了できる場合があります。

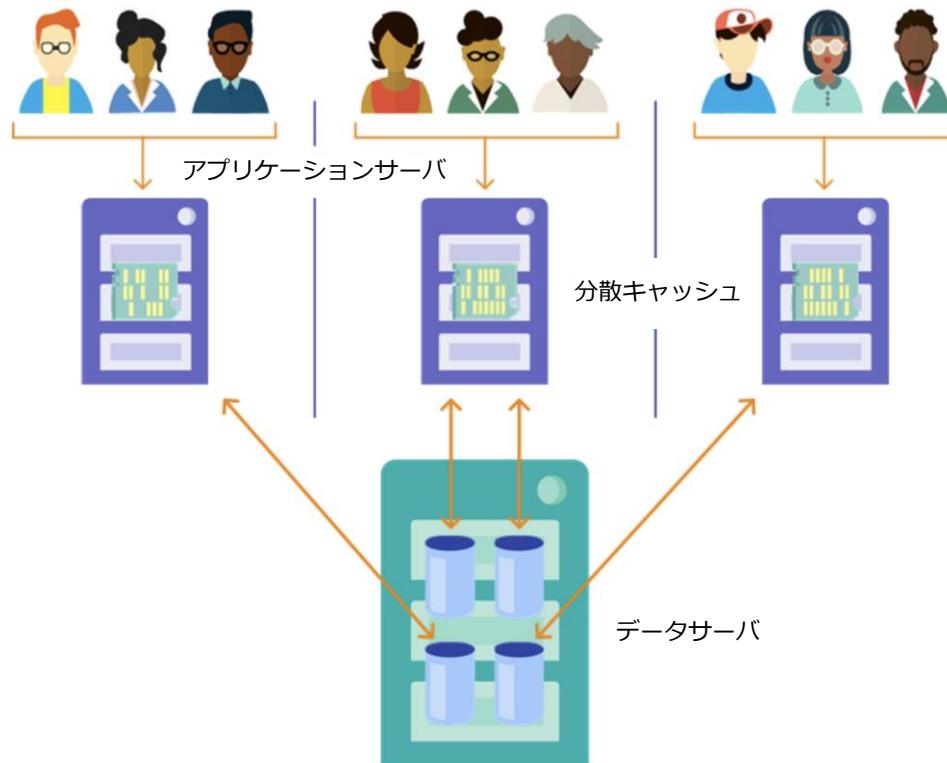
Download mirror database

Previously, adding a database to a mirror required following a manual process copying the database file to the other mirror members, coordinated with API calls to register the database. Starting with this release, the new database can be created automatically on all mirror members and global data is downloaded automatically, without requiring file copies. These improvements significantly simplify the setup of complex mirrored environments, although attention should be paid to the size of the database being added to the mirror, as file copies may still be a practical solution when the database being added already contains a large amount of data. Please refer to the API reference for more details on the new option.



性能/拡張性/セキュリティの強化

ECP管理コマンドを追加



簡単な設定ではあるが、慣れた利用者にとって
コマンドだけで実施できない点がストレス
管理ポータルを必要とせずにECPの設定が可能に
拠点間で構成している場合は、ブラウザアクセスの
ポートを開けることが、セキュリティ上の懸念

Command-line ECP management utility

This release introduces a new ^ECP command line utility to perform common ECP management tasks, similar to existing ^SECURITY and ^DATABASE utilities. This means administrators of ECP environments no longer need to rely on the Management Portal for most configuration tasks.

データサーバ側の設定

アプリケーションサーバ側の設定

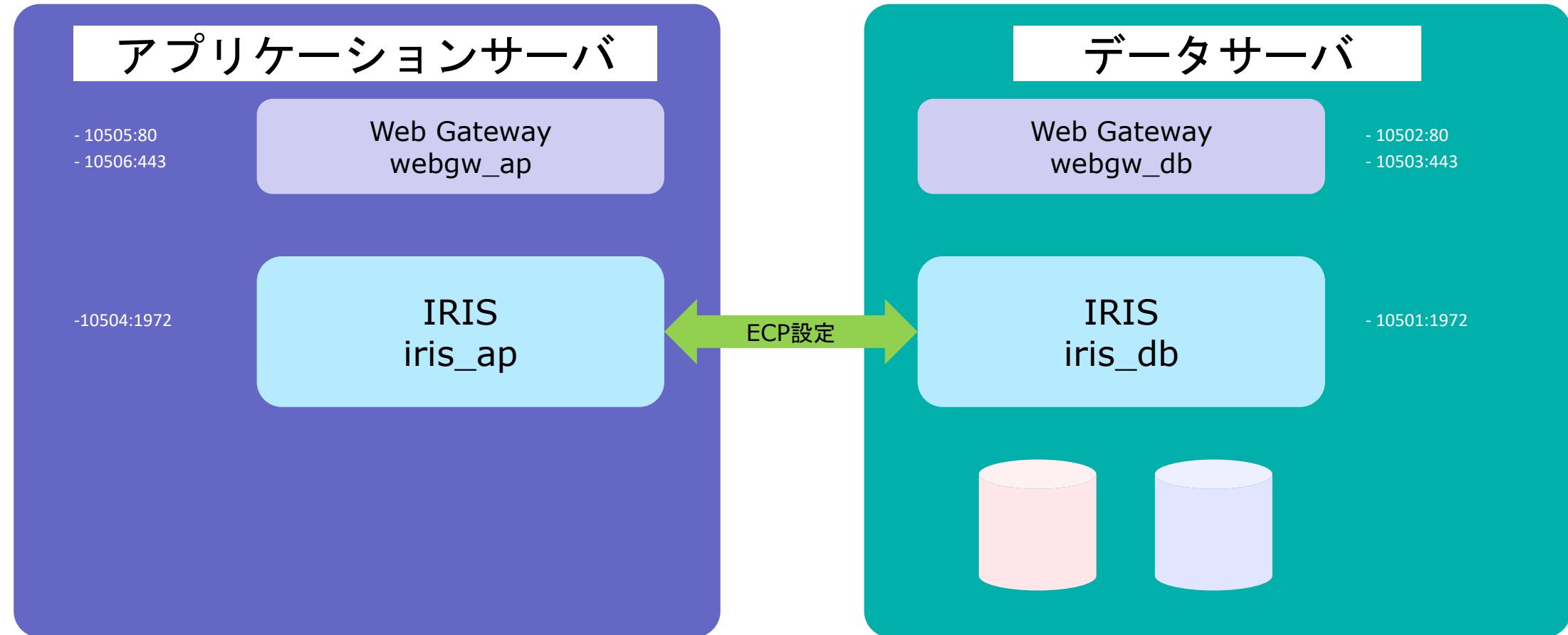




ECPユーティリティ(^ECP)デモ環境構成図：コンテナ構成



参考：<https://github.com/Tomo0Okuyama/iris-2025.1-ecp>
^ECPコマンドを実験できるコンテナ構成の構築





参考：^ECP実行画面（データサーバ側設定）

%SYS>**do ^ECP**

- 1) The system as a data server ←
- 2) The system as an application server

Option? **1**

%Service_ECP is required but currently not enabled
Please use 5) Manage ECP service to enable this service

- 1) List ECP application servers connected to this server
- 2) Manage authorized SSL/TLS Computer Names for this server
- 3) Authorize/Reject pending ECP SSL/TLS connection requests
- 4) Edit ECP settings
- 5) Manage ECP service ←

Option? **5**

Editing %Service_ECP...

Enabled? No => **YES**

Edit list of Allowed Incoming Connections? No => No
Confirm changes to service %Service_ECP? Yes => Yes
Service %Service_ECP updated

- 1) List ECP application servers connected to this server
- 2) Manage authorized SSL/TLS Computer Names for this server
- 3) Authorize/Reject pending ECP SSL/TLS connection requests
- 4) Edit ECP settings
- 5) Manage ECP service

Option? **4**

Max number of application servers: 1 =>
Time interval for troubled state: 60 =>

ECP SSL/TLS support

- 1) Disabled
- 2) Enabled
- 3) Required

ECP SSL/TLS support: 1 =>
Allow remote jobs? Yes =>
Confirm changes to ECP settings? Yes =>
ECP settings updated

- 1) List ECP application servers connected to this server
- 2) Manage authorized SSL/TLS Computer Names for this server
- 3) Authorize/Reject pending ECP SSL/TLS connection requests
- 4) Edit ECP settings
- 5) Manage ECP service

Option? **1**

Client Name	Status	IP Address	IP Port
No results	-----	-----	-----



参考：^ECP実行画面（アプリケーションサーバ側設定）

%SYS>do ^ECP

- 1) The system as a data server
- 2) The system as an application server

Option? **2**

The following is a list of ECP data servers that this system can connect to:

Server Name	Host Name/Port	Status
-------------	----------------	--------

No results

- 1) Add data server
- 2) Edit data server
- 3) List data servers
- 4) Detailed list data servers
- 5) Delete data server
- 6) Configure remote databases
- 7) Edit ECP settings
- 8) Change connection status of data server

Option? **1**

Server Name: **iris_db**

Host DNS Name or IP address: **iris_db**

IP Port number: 1972 =>

Mirror Connection

- | |
|---|
| 1) Non-mirrored connection |
| 2) Mirrored connection |
| 3) Mirrored connection as designated member |

Mirror Connection? 1 =>

Batch Mode? No =>

Use SSL/TLS? No =>

Confirm configuration of data server IRIS_DB? Yes =>

Data server IRIS_DB added

Do you want to establish a connection to it? Yes =>

Connection status changed to: Normal

- 1) Add data server
- 2) Edit data server
- 3) List data servers
- 4) Detailed list data servers
- 5) Delete data server
- 6) Configure remote databases
- 7) Edit ECP settings
- 8) Change connection status of data server

Option? **3**

Server Name	Host Name/Port	Status
-------------	----------------	--------

IRIS_DB	iris_db/1972	Normal
---------	--------------	--------

- 1) Add data server
- 2) Edit data server
- 3) List data servers
- 4) Detailed list data servers
- 5) Delete data server
- 6) Configure remote databases
- 7) Edit ECP settings
- 8) Change connection status of data server

Option? **4**

Server Name:	IRIS_DB
Host Name:	iris_db
IP Port:	1972
Status:	Normal
Mirror Connection:	No
SSL/TLS:	No
Batch Mode:	No

----- 次ページのリモートデータベースの作成に続く -----



参考：^ECP実行画面（アプリケーションサーバ側設定）

リモートデータベースの作成

----- 前ページからの続き -----

- 1) Add data server
- 2) Edit data server
- 3) List data servers
- 4) Detailed list data servers
- 5) Delete data server
- 6) Configure remote databases ←
- 7) Edit ECP settings
- 8) Change connection status of data server

Option? **6**

- 1) Create remote database ←
- 2) Edit remote database
- 3) List remote databases
- 4) Delete remote database

Option? **1**

1) IRIS_DB

Data Server? **1** IRIS_DB
 Remote Directory? **/usr/irissys/mgr/user**
 Database Name: **USERDB**
 Stream Location (optional):
 Confirm creation of remote database USERDB? Yes =>
 Remote database USERDB created

- 1) Create remote database
- 2) Edit remote database
- 3) List remote databases
- 4) Delete remote database

Option? **3**

Database Name	Server Name	Directory
USERDB	IRIS_DB	/usr/irissys/mgr/user/

その後、ネームスペースの作成を行います。

IRISターミナル

- 1) Create remote database
- 2) Edit remote database
- 3) List remote databases ←
- 4) Delete remote database

Option? **3**

Database Name	Server Name	Directory
USERDB	IRIS_DB	/usr/irissys/mgr/user/

**FIPS 140-3**

InterSystems can now interact with the cryptographic libraries of FIPS 140-3. Previous versions supported FIPS 140-2, so this is an important upgrade for users that require the latest FIPS-compliant libraries.

性能/拡張性/セキュリティの強化

FIPS 140-3の対応

FIPS 140-3は、暗号モジュールの設計・実装において高いセキュリティ基準を設定し、政府機関や重要インフラでの利用を想定しています。この規格に準拠することで、暗号モジュールの信頼性と安全性が保証され、機密データの保護が強化されます。日本を含む多くの国でもこの規格を参照し、セキュリティ要件の基準としています。

FIPS140-3に対応することによるエンドユーザにとってのメリット

国際標準への適合

- 国際的セキュリティ基準への準拠。グローバルビジネス展開時の優位性。
- 金融・医療分野など規制産業でのコンプライアンス負担軽減。

セキュリティ強化とリスク低減

- 高度な暗号化技術による堅牢なデータ保護。
- サイバー攻撃対策の強化。情報漏洩リスクの縮小。
- 患者情報や金融データなど機密情報の安全性向上。

ビジネス面での優位性

- 高セキュリティ基準による顧客・取引先からの信頼獲得。
- 政府調達案件への参画機会の増加。
- 海外企業とのパートナーシップ構築の円滑化。

運用効率の最適化

- 標準化されたセキュリティフレームワークによるコスト効率向上。
- セキュリティ監査プロセスの効率化。



新しいインターフェース【Production Configuration】

プロダクション構成

The screenshot shows the InterSystems IRIS Management Portal with the Production Configuration module open. The interface includes a top navigation bar with tabs like 'ホーム', '概要', 'ヘルプ', and 'ログアウト'. On the left, there's a sidebar with sections for 'サービス' (OrderRESTService, OrderProcessor), 'プロセス' (EmailOperation, InventoryOperation, OrderStorageOperation), and 'オペレーション' (Production Settings, Operations, Operation). A large central area displays the 'INTEROPERABILITY' configuration for 'Production PRDDEMO.OrderProcessingProduction'. It shows a flowchart with 'Inbound Hosts' (OrderRESTService) connected to a 'Process Hosts' node (OrderProcessor), which then connects to 'Outbound Hosts' (EmailOperation, InventoryOperation, OrderStorageOperation). The right side of the screen contains detailed configuration panels for the OrderProcessor host, including sections for 'Properties', 'Informational Settings', 'Comment', 'Category', 'Class Name', 'Description', 'Adapter Class Name', 'Adapter Description', 'Business Partner', and various 'Settings' and 'Control' sections.

[Try the new UI]ボタン押下で
新しいインターフェースが利用できる

現在社内外からのフィードバックを収集し開発進行中。2025.2で大幅改修を予定



新しいインターフェース【DTL Editor】

データ変換

The screenshot illustrates the transition from the old DTL Editor interface to the new one. On the left, the 'Try the new UI' button is highlighted with a red box. The right side shows the updated interface with a more modern design, including a navigation bar with 'Back to standard UI', 'New', 'Open', 'Save', 'Save As', 'Compile', and 'Test' buttons.

Actions (0 selected)

	Action Type	Action Details	Description
1	SET	target.patientId = ##class(UHR.Util.DTLFunctions).FormatUPI(source.patientNumber)	病院Aの患者番号を統一患者IDに変換
2	SET	target.personalDetails.lastName = source.personalInfo.lastName	
3	SET	target.personalDetails.firstName = source.personalInfo.firstName	
4	SET	target.personalDetails.dateOfBirth = ##class(UHR.Util.DTLFunctions).FormatPDate(source.personalInfo.dob)	
5	SWITCH		性別コードを標準形式に変換
6	CASE	source.personalInfo.gender = "M"	
7	SET	target.personalDetails.gender = male	
8	CASE	source.personalInfo.gender = "F"	
9	SET	target.personalDetails.gender = female	
10	DEFAULT		
11	SET	target.personalDetails.gender = unknown	
12	SET	target.personalDetails.contact.phoneNumber = source.personalInfo.contactDetails.phone	

[Try the new UI]ボタン押下で
新しいインターフェースが利用できる

現在社内外からのフィードバックを収集し開発進行中。2025.2で大幅改修を予定



新しいDTL Editorインターフェース：横レイアウト

DTL EDITOR

DTLDEMO.HospitalAtoUHR

Hospital.A.PatientData Create new **SET** UHR.Patient

Actions (4 selected)

Select all **X** **Disable** **New**

	Action	Description
<input type="checkbox"/> 1	SET	target.patientId = ##class(UHR.Util.DTLFunctions).FormatUPI(source.patientNumber)
<input type="checkbox"/> 2	SET	target.personalDetails.lastName = source.personalInfo.lastName
<input type="checkbox"/> 3	SET	target.personalDetails.firstName = source.personalInfo.firstName
<input type="checkbox"/> 4	SET	target.personalDetails.dateOfBirth = ##class(UHR.Util.DTLFunctions).FormatJDate(source.personalInfo.dob)
<input type="checkbox"/> 5	SWITCH	性別コードを標準形式に変換
<input type="checkbox"/> 6	CASE	source.personalInfo.gender = "M"
<input type="checkbox"/> 7	SET	target.personalDetails.gender = male
<input type="checkbox"/> 8	CASE	source.personalInfo.gender = "F"
<input type="checkbox"/> 9	SET	target.personalDetails.gender = female
<input type="checkbox"/> 10	DEFAULT	
<input type="checkbox"/> 11	SET	target.personalDetails.gender = unknown
<input type="checkbox"/> 12	SET	target.personalDetails.contact.phoneNumber = source.personalInfo.contactDetails.phone
<input type="checkbox"/> 13	SET	target.personalDetails.contact.email = source.personalInfo.contactDetails.email
<input type="checkbox"/> 14	SET	target.personalDetails.contact.address = source.personalInfo.contactDetails.address
<input type="checkbox"/> 15	SET	target.insurance.provider = source.insuranceInfo.provider
<input type="checkbox"/> 16	SET	target.insurance.policyNumber = source.insuranceInfo.policyNumber
<input type="checkbox"/> 17	SET	target.insurance.validUntil = ##class(UHR.Util.DTLFunctions).FormatJDate(source.insuranceInfo.validUntil)
<input type="checkbox"/> 18	SET	target.metadata.lastUpdated = ##class(UHR.Util.DTLFunctions).FormatDateTime(source.lastUpdated)
<input type="checkbox"/> 19	SET	target.metadata.originalId = source.patientNumber

source

- patientNumber
- personalInfo
- lastName
- firstName
- dob
- gender
- contactDetails
- insuranceInfo
- provider
- policyNumber
- validUntil
- medicalHistory()
- visitDate
- department
- doctorName
- symptoms
- diagnosis
- medication()
- lastUpdated

target

- patientId
- personalDetails
- lastName
- firstName
- dateOfBirth
- gender
- contact
- insurance
- provider
- policyNumber
- validUntil
- encounters()
- encounterDate
- clinician
- department
- symptoms
- diagnosis
- medications()
- metadata
- sourceSystem
- originalId
- lastUpdated

アップグレードチェックリスト



InterSystems Upgrade Impact Checklist

This form helps you assess the impact of upgrading to a new version of your InterSystems product. To see a list of changes that may impact the code, configuration, or operation of your system after the upgrade, simply select your product and enter the versions from which and to which you are upgrading. The relevant changes will appear below.

From time to time, InterSystems publishes items of immediate importance to users of our software. These include alerts, mission-critical issues, important upgrades, fixes, and release announcements. You can obtain the most current list at <https://www.intersystems.com/support-learning/support/product-news-alerts/>. InterSystems recommends that you subscribe to receive notifications about support alerts, critical issues, fixes and product releases at <https://www.intersystems.com/support/product-alerts-advisories>.

Data Platform Instance Details

Product	Version From	Version To
InterSystems IRIS Data Platform	2024.1.3	2025.1.0

Data Platform Features Used

Select/Deselect All

Core Features	Cloud Operations
<input checked="" type="checkbox"/> Core	<input type="checkbox"/> Cloud
<input type="checkbox"/> Security	<input type="checkbox"/> Interoperability
Analytics	<input type="checkbox"/> Interoperability
<input type="checkbox"/> Business Intelligence	Scalability
<input type="checkbox"/> Machine Learning	<input checked="" type="checkbox"/> Mirroring
<input type="checkbox"/> Natural Language Processing	<input type="checkbox"/> Sharding
Application Development	Data Management
<input type="checkbox"/> Development Tools	<input type="checkbox"/> SQL
<input type="checkbox"/> Driver Technologies	Other
<input type="checkbox"/> Embedded Python	<input type="checkbox"/> Monitoring
<input type="checkbox"/> External Languages	
<input type="checkbox"/> Web Applications	

Generate Upgrade Checklist

© 2024 InterSystems Corporation, Cambridge, MA. All rights reserved.

[Privacy & Terms](#) [Guarantee](#) [Accessibility](#)
[Cookies Settings](#) [Cookie List](#)

<https://docs.intersystems.com/upgrade/>

Upgrade Impact Checklist from 2024.3.0 to 2025.1.0

Go Back Export as CSV Audience: All Features:

Categories Core: 10 Interoperability: 3

Core

DP-420686: Turn on compact IEEE in server by default Administrator
2025.1.0

Previous versions introduced the ability to use compact IEEE values when they are stored in a \$LIST. This change makes the compact behavior the default for new installations by setting the ListFormat configuration parameter in the [Miscellaneous] section of the CPF to 1. However, old versions of database drivers (such as JDBC, ODBC, and ADO.NET) will be unable to connect with any server that uses compact IEEE doubles. To avoid connection failures, update your database drivers to the latest version or turn compact IEEE doubles off by manually setting the ListFormat configuration parameter in the [Miscellaneous] section of the CPF to 0.

DP-421228: Change journal format, reducing the sizes of records Administrator
2025.1.0

This change reduces the size of journal records, which speeds journaling. With this change, you cannot restore journal files created in an instance running 2025.1.0 or later on an instance running any earlier version. As a consequence, for a mirrored system, when you upgrade to a version from one version to another that uses a different journal file format, you must take care to do the upgrades in the

アップグレードを実施すると、どのように動作が変化するかを確認できるチェックリストです。既存環境のバージョンと、アップグレード先のバージョンを指定し、そのバージョン間での仕様変更を確認できます。アップグレードを検討する際に活用してください



参考：2024.3から2025.1への変更点：2025/04/23時点

主な変更点の概要（日本語訳）

1. コンパクトIEEEのデフォルト有効化 (DP-420686)
 - ・ 新規インストールでは、コンパクトIEEE値をデフォルトで使用 (ListFormat=1)。
 - ・ 古いデータベースドライバ (JDBC、ODBC、ADO.NET) では接続不可。最新ドライバへの更新またはListFormat=0で無効化が必要。
2. ジャーナル形式の変更 (DP-421228)
 - ・ ジャーナルレコードのサイズを縮小し、高速化。
 - ・ 2025.1.0以降のジャーナルは旧バージョンで復元不可。ミラーシステムではバックアップを先にアップグレード必須。
3. Type^JRNDUMPの削除 (DP-431508)
 - ・ Type^JRNDUMPを廃止。代替として##class(%JournalRecordType).LogicalToDisplay()を使用。
4. プロセスごとのデータベースファイルテーブルの拡張 (DP-434735)
 - ・ 1,000以上のI/Oファイルを開く場合、OSの制限を $5 + \text{MIN}(x^2, 10240 + x)$ に設定必要 (Windows除外)。
5. \$SYSTEM.SQL.ParseQualifiers()の改善 (DP-435416)
 - ・ 空文字列やフラグのみのレガシークオリファイアを受け入れ、エラー処理を強化。
6. 超越関数の丸め処理変更 (DP-435863)
 - ・ Windowsでの三角関数 (sin、cosなど) の丸め方をLinuxと一致するよう調整。
7. 引数の新挙動 (DP-437710)
 - ・ method(x...)で整数値xを渡すと、x個の省略引数をシミュレート。非整数は無視、超過時はエラー。
8. %Net.SMTPのログ機能強化 (DP-437821)
 - ・ ログを^ISCLOGに保存。ログレベル (1:無効、2:認証のみ、3:全通信) を設定可能。
9. グローバル設定の非推奨 (DP-437861)
 - ・ New Global Growth/Pointer Block設定を非推奨。管理ポータルから非表示化。
10. CompileAllNamespaces()の挙動統一 (DP-437953)
 - ・ 全ての名前空間に同一のqspecを適用 (従来は名前空間ごとに微調整)。
11. Ens.DataType.UTCのエラー処理強化 (DP-434849)
 - ・ 無効な日付 (例: 6月31日) で一貫してILLEGAL VALUEエラーを発生。
12. バッチ再送信時のエラー (DP-436058)
 - ・ UseDefaultBatch有効時、再送信でエラー発生。受付にはAcceptDefaultBatchParentBatchを有効化。
13. FTPアダプターの改善 (DP-436276)
 - ・ GetSizeエラーでスキップせず再試行可能。新設定でエラー種別追加可。

2024.3リリースについて



公式ドキュメント:2024.3リリースノート(英文)

The screenshot shows the 'New in InterSystems IRIS 2024.3' section of the documentation. It includes a sidebar with navigation links like 'Release Notes' (highlighted in red) and 'Known Issues and Notes'. The main content area discusses new features such as 'Ability to Resend Messages from Visual Trace', 'Enhanced Rule Editor Capabilities', and 'Enhancing Analytics and AI'. A URL at the bottom is provided for the full document.

https://docs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GCRN_new20243

製品のリリースサイクル



開発者向け機能強化

- Visual Traceからのメッセージ再送機能: 「再送」ボタンによりメッセージフロー修正が簡素化
- ルールエディターの拡張: フィルター機能、自動スクロール、大規模ルールセット対応の改善

AIおよびアナリティクスの機能強化

- ベクトル検索: 実験的機能から本番環境対応へ移行し、検索速度が3~4倍向上
- EMBEDDINGデータ型: テキストからベクトル埋め込みを自動生成
- Adaptive Analytics: 柔軟な出力フォーマット、追加MDX関数、Excelのカスタムグループ機能

SQLおよびデータ管理の強化

- SQLクエリプラン: クエリ改善のための情報提供と警告メッセージ
- DMLステートメント用CTE: 複雑なSQLステートメントの可読性向上

性能/拡張性/セキュリティの強化

- 選択的SQL監査: ステートメントタイプに基づく詳細な監査ポリシー
- データベース拡張の高速化: Linux/Unixで50~100倍の速度向上
- テーブルシャード変換: 稼働中のデータでもシャード/非シャード間の変換が可能



開発者向け機能強化

Visual Traceからメッセージの再送が可能に

この機能はVisual Traceに簡単な「再送信」ボタンを導入し、ユーザーが複雑なメッセージ検索を行わずにトレースから直接メッセージを再送信できるようにします。これにより、統合シナリオでのメッセージフローの修正に必要な時間が大幅に短縮されます。

2024.2

VISUAL TRACE

Session ID: 8 [Legend](#) [Printable Version](#) Go to items 1 - 3 Items per page 40 Show events Show internal items Apply Filter None Previous Page

Services	Processes	Operations
FileService	FileRouter	WritePDFsOp
[1] 2024-12-17 09:54:26.031 StreamContainer		
	[2]	
	[3] 2024-12-17 09:54:26.033 StreamContainer	

Header Body Contents

<ObjectId>	8
SessionId	8
TargetConfigName	FileRouter
SourceConfigName	FileService
MessageBodyClassName	Ens.StreamContainer
MessageBodyId	3
TimeCreated	2024-12-17 09:54:26.031
TimeProcessed	2024-12-17 09:54:26.034
Type	Request
Priority	Async
SourceBusinessType	BusinessService
TargetQueueName	FileRouter
TargetBusinessType	BusinessProcess
ReturnQueueName	
CorrespondingMessageId	
BusinessProcessId	
Description	
Invocation	Queue
SuperSession	
Resent	
Status	Completed
Is Error	0
Error Status	OK
Banked	0

2024.3

VISUAL TRACE

Session ID: 4 [Legend](#) [Printable Version](#) Go to items 1 - 3 Items per page 40 Show events Show internal items Apply Filter None Previous Page

Services	Processes	Operations
FileService	FileRouter	WritePDFsOp
[1] 2024-12-17 10:10:49.875 StreamContainer		
	[2]	
	[3] 2024-12-17 10:10:49.877 StreamContainer	

Header Body Contents

<ObjectId>	4
Session Id	4
TargetConfigName	FileRouter
SourceConfigName	FileService
MessageBodyClassName	Ens.StreamContainer
MessageBodyId	2
TimeCreated	2024-12-17 10:10:49.875
TimeProcessed	2024-12-17 10:10:49.878
Type	Request
Priority	Async
SourceBusinessType	BusinessService
TargetQueueName	FileRouter
TargetBusinessType	BusinessProcess
ReturnQueueName	
CorrespondingMessageId	
BusinessProcessId	
Description	
Invocation	Queue
SuperSession	
Resent	
Status	Completed
Is Error?	0
Error Status	OK
Banked	0

Resend



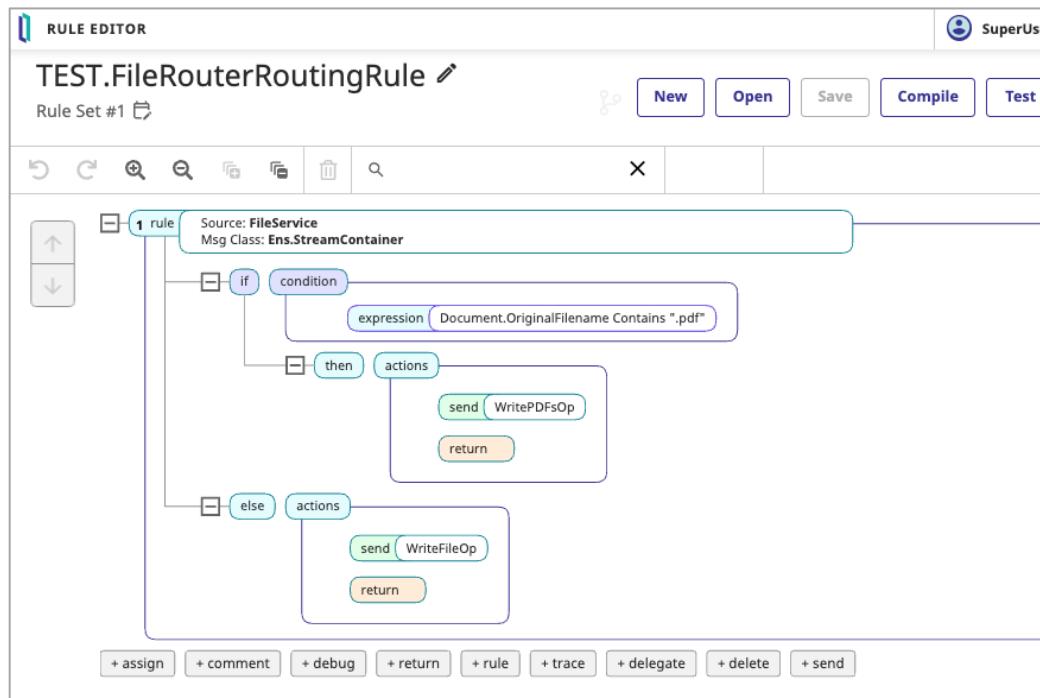
開発者向け機能強化

ルールエディタの拡張

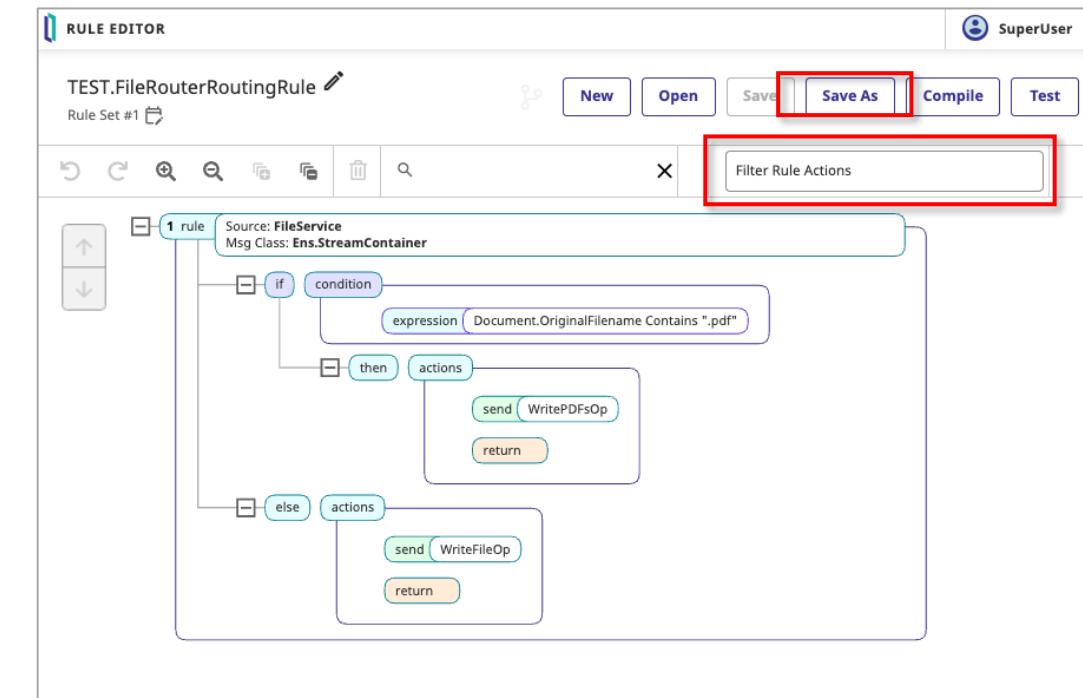
ルールエディタの機能強化により、操作性と拡張性が向上しました。

- フィルタにルールを指定:
 - ルールへの自動スクロール:
 - 大規模なルールの編集を使いやすく:
- ルールの確認と編集をしやすくするツールバー「フィルタ」を追加
エディター内で指定したルールに自動スクロール
大規模なルールの作成や編集をしやすくするようインターフェースを改善

2024.2



2024.3





開発者向け機能強化

ルールエディタの拡張

2024.2

RULE EDITOR

TEST.FileRouterRoutingRule

Rule Set #1

New Open Save Save As Compile

Comment double-click to edit comment

+ if condition

+ else actions

comment double-click to edit comment

+ assign + comment + debug + return + rule + trace + delegate + delete + send

2024.3

RULE EDITOR

TEST.FileRouterRoutingRule

Rule Set #1

New Open Save Save As Compile

Comment double-click to edit comment

+ if condition

+ else actions

+ condition + comment

+ assign + comment + debug + return + rule + trace + delegate + delete + send

comment double-click to edit comment



SQLおよびデータ管理の強化

SQLクエリプランの機能強化

新しいSQLクエリプランには、クエリ構造、スキーマ、またはシステム設定の改善についてユーザーに知らせるための情報メッセージと警告メッセージが含まれるようになりました。

2024.2

実行 プラン表示 履歴を表示 クエリビルダ 表示モード 最大 1000 その他オプション

```
SELECT * FROM test_no_stats
WHERE value > 500 AND name LIKE 'Name1%'
```

実行プランが以下に表示されます:

ステートメント・テキスト

```
SELECT * FROM test_no_stats WHERE value > ? AND name LIKE ? /*#OPTIONS
{"DynamicSQLTypeList":"10,1"} */
```

クエリプラン

相対コスト = 1307000

- Read master map SQLUser.test_no_stats.IDKEY, looping on ID1.
- For each row:
 - Test the %PATTERN condition on %SQLUPPER(name), the %STARTSWITH condition on %SQLUPPER(name), the > condition on value, and the NOT NULL condition on %SQLUPPER(name).
 - Output the row.

2024.3

実行 プラン表示 履歴を表示 クエリビルダ 表示モード 最大 1000 その他オプション

```
SELECT * FROM test_no_stats
WHERE value > 500 AND name LIKE 'Name1%'
```

実行プランが以下に表示されます:

ステートメント・テキスト

```
SELECT * FROM test_no_stats WHERE value > ? AND name LIKE ? /*#OPTIONS
{"DynamicSQLTypeList":"10,1"} */
```

警告

- Table SQLUser.test_no_stats is not tuned.

クエリプラン

相対コスト = 1307000

- Read master map SQLUser.test_no_stats.IDKEY, looping on ID1.
- For each row:
 - Test the %PATTERN condition on %SQLUPPER(name), the %STARTSWITH condition on %SQLUPPER(name), the > condition on value, and the NOT NULL condition on %SQLUPPER(name).
 - Output the row.

```
CREATE TABLE test_no_stats (
  id INT PRIMARY KEY,
  name VARCHAR(50),
  value NUMERIC(10,2)
)
```

id	name	value
1	Name1	10.50
2	Name2	21.00
3	Name3	31.50

3 row(s) affected



SQLおよびデータ管理の強化

SQLクエリプランの機能強化

テーブルチューニングが未実施であり、性能改善の余地がある警告を表示

2024.2

実行 プラン表示 履歴を表示 クエリビルダ 表示モード 最大 1000 その他オプション

```
SELECT * FROM test_index_selectivity
WHERE status = 'ACTIVE'
```

実行プランが以下に表示されます:

ステートメント・テキスト

```
SELECT * FROM test_index_selectivity WHERE status = ? /*#OPTIONS
{"DynamicSQLTypeList":"1"} */
```

クエリプラン

相対コスト = 84400

- Read index map SQLUser.test_index_selectivity.idx_status, using the given %SQLUPPER(status), and looping on ID1.
- For each row:
 - Read master map SQLUser.test_index_selectivity.IDKEY, using the given idkey value.
 - Output the row.

```
<plans> <plan> <sql> SELECT * FROM test_index_selectivity WHERE status = ?
/*#OPTIONS {"DynamicSQL":1} */ /*#OPTIONS {"DynamicSQLTypeList":"1"} */ </sql>
<cost value="84400"/> Read index map SQLUser.test_index_selectivity.idx_status,
using the given %SQLUPPER(status), and looping on ID1. For each row: Read master
map SQLUser.test_index_selectivity.IDKEY, using the given idkey value. Output the row.
</plan> </plans>
```

```
CREATE TABLE test_index_selectivity (
    id INT PRIMARY KEY,
    status VARCHAR(10),
    data VARCHAR(50)
)
```

id	status	data
1	ACTIVE	Data 1
2	ACTIVE	Data 2
3	ACTIVE	Data 3
4	ACTIVE	Data 4
5	INACTIVE	Data 5

5 row(s) affected

CREATE INDEX idx_status
ON test_index_selectivity(status)

2024.3

実行 プラン表示 履歴を表示 クエリビルダ 表示モード 最大 1000 その他オプション

```
SELECT * FROM test_index_selectivity
WHERE status = 'ACTIVE'
```

実行プランが以下に表示されます:

ステートメント・テキスト

```
SELECT * FROM test_index_selectivity WHERE status = ? /*#OPTIONS {"DynamicSQLTypeList":"1"} */
```

警告

- Table SQLUser.test_index_selectivity is not tuned.

クエリプラン

相対コスト = 84400

- Read index map SQLUser.test_index_selectivity.idx_status, using the given %SQLUPPER(status), and looping on ID1.
- For each row:
 - Read master map SQLUser.test_index_selectivity.IDKEY, using the given idkey value.
 - Output the row.

```
<plans> <plan> <sql> SELECT * FROM test_index_selectivity WHERE status = ?
/*#OPTIONS {"DynamicSQL":1} */ /*#OPTIONS {"DynamicSQLTypeList":"1"} */ </sql>
<warning> Table SQLUser.test_index_selectivity is not tuned. </warning> <cost
value="84400"/> Read index map SQLUser.test_index_selectivity.idx_status, using the
given %SQLUPPER(status), and looping on ID1. For each row: Read master map
SQLUser.test_index_selectivity.IDKEY, using the given idkey value. Output the row.
</plan> </plans>
```



SQLおよびデータ管理の強化

SQLクエリプランの機能強化

テーブルチューニングが未実施であり、性能改善の余地がある警告を表示

2024.2

実行 プラン表示 履歴を表示 クエリビルダ 表示モード ✓ 最大 1000 その他オプション

```
SELECT a.*, b.*  
FROM test_collation1 a  
JOIN test_collation2 b ON a.name = b.name
```

実行プランが以下に表示されます:

ステートメント・テキスト

```
SELECT a.* , b.* FROM test_collation1 a JOIN test_collation2 b ON a.name = b.name
```

クエリプラン

相対コスト = 140341744

- Read master map SQLUser.test_collation1(A).IDKEY, looping on A.ID1.
- For each row:
 - [Call module C](#) once, which populates temp-file A.
 - Read temp-file A, using the given B.name, and looping on B.ID1.
 - For each row:
 - Output the row.

Module: C

- Read master map SQLUser.test_collation2(B).IDKEY, looping on B.ID1.
- For each row:
 - Test the NOT NULL condition on B.name.
 - Add a row to temp-file A, subscripted by B.name and B.ID1, with node data of B.id.

2024.3

実行 プラン表示 履歴を表示 クエリビルダ 表示モード ✓ 最大 1000 その他オプション

```
SELECT a.* , b.*  
FROM test_collation1 a  
JOIN test_collation2 b ON a.name = b.name
```

実行プランが以下に表示されます:

ステートメント・テキスト

```
SELECT a.* , b.* FROM test_collation1 a JOIN test_collation2 b ON a.name = b.name
```

警告

- Tables SQLUser.test_collation1 and SQLUser.test_collation2 are not tuned.

クエリプラン

相対コスト = 140341744

- Read master map SQLUser.test_collation1(A).IDKEY, looping on A.ID1.
- For each row:
 - [Call module C](#) once, which populates temp-file A.
 - Read temp-file A, using the given B.name, and looping on B.ID1.
 - For each row:
 - Output the row.

Module: C

- Read master map SQLUser.test_collation2(B).IDKEY, looping on B.ID1.
- For each row:
 - Test the NOT NULL condition on B.name.
 - Add a row to temp-file A, subscripted by B.name and B.ID1, with node data of B.id.

id	name
1	Test
2	test

2 row(s) affected

id	name
1	Test
2	test

2 row(s) affected



SQLおよびデータ管理の強化

CTE(Common Table Expression)でDML文をサポート

文の先頭で共通テーブル式 (CTE) を定義するために使用されるWITH句が、DML文でも使用できるようになりました。これにより、複雑なINSERT ... SELECT文や、複雑なFROM句を持つUPDATEおよびDELETE文を、事前にCTE定義を使って簡素化できるようになります。文全体の可読性が向上します。

右の実行例では、顧客マスタ、返品情報、売上情報の3つから、購入金額が100万円以上かつ返品がない顧客にVIPフラグを付ける (UPDATE) という処理をWITH句を利用して行います。

【次ページに具体的なテーブル情報が記載されています】

2024.3 以降で実行可能

```

WITH HighValueCustomers AS (
    -- 100万円超の顧客を抽出
    SELECT CustomerID
    FROM Sales
    WHERE YEAR(SaleDate) = 2023
    GROUP BY CustomerID
    HAVING SUM(Amount) > 1000000
),
NoReturns AS (
    -- 返品がない顧客を抽出
    SELECT DISTINCT CustomerID
    FROM Customer c
    WHERE NOT EXISTS (
        SELECT 1
        FROM Returns r
        WHERE r.CustomerID = c.CustomerID
        AND YEAR(ReturnDate) >= 2021
    )
)
UPDATE Customer
SET Status = 'VIP'
WHERE CustomerID IN (
    -- 両方の条件を満たす顧客を特定
    SELECT h.CustomerID
    FROM HighValueCustomers h
    INNER JOIN NoReturns n ON h.CustomerID = n.CustomerID
)

```



SQLおよびデータ管理の強化

CTE(Common Table Expression)でDML文をサポート

-- 顧客マスタ

```
CREATE TABLE Customer (
    CustomerID INT PRIMARY KEY,
    Name VARCHAR(100) NOT NULL,
    Status VARCHAR(20) DEFAULT 'Regular', -- Regular, VIP などのステータス
    Email VARCHAR(255),
    Phone VARCHAR(20),
    CreatedDate DATETIME DEFAULT CURRENT_TIMESTAMP,
    UpdatedDate DATETIME
)
```

CustomerID	Name	Status	Email	Phone	CreatedDate	UpdatedDate
1	山田太郎	Regular	yamada@example.com	090-1111-2222	2025-01-06 11:26:29	
2	鈴木花子	Regular	suzuki@example.com	090-2222-3333	2025-01-06 11:25:19	
3	佐藤健一	Regular	sato@example.com	090-3333-4444	2025-01-06 11:25:38	
4	田中美咲	Regular	tanaka@example.com	090-4444-5555	2025-01-06 11:25:48	
5	渡辺順二	Regular	watanabe@example.com	090-5555-6666	2025-01-06 11:25:57	

5 row(s) affected

-- 返品情報

```
CREATE TABLE Returns (
    ReturnID INT PRIMARY KEY,
    CustomerID INT NOT NULL,
    SaleID INT NOT NULL,
    ReturnDate DATETIME NOT NULL,
    Reason VARCHAR(500),
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (SaleID) REFERENCES Sales(SaleID)
)
```

ReturnID	CustomerID	SaleID	ReturnDate	Reason
1	2	4	2023-02-10 00:00:00	商品不良
2	5	13	2023-02-15 00:00:00	サイズ不適合
3	2	5	2023-06-20 00:00:00	注文間違い

3 row(s) affected

-- 売上情報

```
CREATE TABLE Sales (
    SaleID INT PRIMARY KEY,
    CustomerID INT NOT NULL,
    SaleDate DATETIME NOT NULL,
    Amount DECIMAL(12,2) NOT NULL, -- 売上金額/購入金額
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
)
```

SaleID	CustomerID	SaleDate	Amount
1	1	2023-01-15 00:00:00	300000.00
2	1	2023-04-20 00:00:00	400000.00
3	1	2023-08-10 00:00:00	500000.00
4	2	2023-02-05 00:00:00	200000.00
5	2	2023-06-15 00:00:00	300000.00
6	2	2023-10-25 00:00:00	300000.00
7	3	2023-03-01 00:00:00	600000.00
8	3	2023-07-12 00:00:00	400000.00
9	3	2023-11-30 00:00:00	500000.00
10	4	2023-05-08 00:00:00	400000.00
11	4	2023-09-18 00:00:00	300000.00
12	4	2023-12-20 00:00:00	400000.00
13	5	2023-01-30 00:00:00	200000.00
14	5	2023-06-25 00:00:00	200000.00
15	5	2023-12-10 00:00:00	200000.00

15 row(s) affected



SQLおよびデータ管理の強化

CTE(Common Table Expression)でDML文をサポート

2024.2

```
実行 プラン表示 履歴を表示 クエリビルダ 表示モード ▾ 最大 1000 その他オプション

WITH HighValueCustomers AS (
    -- 100万円超の顧客を抽出
    SELECT CustomerID
    FROM Sales
    WHERE YEAR(SaleDate) = 2023
    GROUP BY CustomerID
    HAVING SUM(Amount) > 1000000
),
NoReturns AS (
    -- 返品がない顧客を抽出
    SELECT DISTINCT CustomerID
    FROM Customer c
    WHERE NOT EXISTS (
        SELECT 1
        FROM Returns r
        WHERE r.CustomerID = c.CustomerID
        AND YEAR(ReturnDate) >= 2021
    )
)
UPDATE Customer
SET Status = 'VIP'
WHERE CustomerID IN (
    -- 両方の条件を満たす顧客を特定
    SELECT h.CustomerID
    FROM HighValueCustomers h
    INNER JOIN NoReturns n ON h.CustomerID = n.CustomerID
)
```

このステートメントの作成中に次のエラーが発生しました。

エラー

SQLCODE: -383
Message: CTE is only supported for SELECT statements.

2024.3

```
実行 プラン表示 履歴を表示 クエリビルダ 表示モード ▾ 最大 1000 その他オプション

WITH HighValueCustomers AS (
    -- 100万円超の顧客を抽出
    SELECT CustomerID
    FROM Sales
    WHERE YEAR(SaleDate) = 2023
    GROUP BY CustomerID
    HAVING SUM(Amount) > 1000000
),
NoReturns AS (
    -- 返品がない顧客を抽出
    SELECT DISTINCT CustomerID
    FROM Customer c
    WHERE NOT EXISTS (
        SELECT 1
        FROM Returns r
        WHERE r.CustomerID = c.CustomerID
        AND YEAR(ReturnDate) >= 2021
    )
)
UPDATE Customer
SET Status = 'VIP'
WHERE CustomerID IN (
    -- 両方の条件を満たす顧客を特定
    SELECT h.CustomerID
    FROM HighValueCustomers h
    INNER JOIN NoReturns n ON h.CustomerID = n.CustomerID
)
```

行数: 2 パフォーマンス: 0.0047 秒 1950 グローバル参照 32943 実行されたコマンド 0 ディスク読み込みレイテンシ [ms] クエリ:%sqlcq_DEMO.cls10 最終更新:2025-04-22 17:42:34.550

2 行が影響を受けました



性能/拡張性/セキュリティの強化

データベースとWIJファイルの拡張速度向上

LinuxとUnixプラットフォームにおいて、データベースとWIJ (Write Image Journal) ファイルの拡張速度が大幅に向上しました。

拡張方法の変更

従来は空のブロックを書き込んで拡張していましたが、新バージョンでは
ファイルシステムのメタデータ操作だけで拡張できるようになりました。

速度向上

拡張速度が**50～100倍**も速くなりました

対応ファイルシステム

Linux: XFS、ext4、Btrfs
AIX: JSF2

対応していないファイルシステムでも、拡張コードの最適化により限定的な改善が見られます

WIJ:InterSystems IRISのデータ保護メカニズム

IRISでは「ライトイメージジャーナリング (Write Image Journaling)」と呼ばれる2段階ステップを通じてデータベース更新を保護しています。

仕組み：

1. 起動時のWIJファイル作成:

IRISの起動時に、Write (書き込み) デーモンが「ライトイメージジャーナル (WIJ)」ファイルを作成します

2. 2段階の更新プロセス:

第1段階：データベース更新は最初にメモリから一時的なジャーナルファイル「IRIS.WIJ」に書き込まれます

第2段階：その後、実際のデータベースに書き込まれます

3. システムクラッシュからの回復:

システムが第2段階（データベースへの書き込み）中にクラッシュした場合でも

回復時にWIJファイルから未完了の更新を再適用できます

この仕組みにより、システム障害が発生してもデータの整合性が保たれ、データベースが破損するリスクを大幅に低減しています。



性能/拡張性/セキュリティの強化

SQL監査の機能強化

SQLステートメントに対してより詳細な監査ポリシーを定義できるようになりました。以前から利用可能だったアクセスメカニズム（データベースドライバ、動的SQL、組み込みSQL）に加えて、**ステートメントのタイプに基づいて監査ポリシーを細かく調整できます**。例えば、データを変更するすべてのステートメント（DMLステートメント）を監査する一方で、データを読み取るだけのクエリは監査対象から除外するといったポリシーを設定できるようになりました。

2024.2

監査データベースの閲覧

イベントソース	ページサイズ:	結果:	ページ:
*	1000	9	[1] の 1
イベントタイプ			
*			
イベント名	*		
システムID	*		
PID	*		
ユーザ	* (すべて)		
Admin			
CSPSystem			
IAM			
SuperUser			
UnknownUser			
認証	ハッシュ		
認証なし			
Kerberos			
Kerberos暗号化			
Kerberos/ケケット整合性			
相互TLS承認			
開始日時	<input type="text"/>		
終了日時	<input type="text"/>		

時間

イベントソース イベントタイプ イベント PID CSPセッション ユーザ

2025-04-16 00:43:16.442 %System %Security AuditReport 661 SAIrnWgxW _SYSTEM
 2025-04-16 00:34:09.351 %System %Security AuditChange 661 SAIrnWgxW _SYSTEM
 2025-03-25 01:33:13.106 %System %Security AuditChange 673 _SYSTEM
 2025-03-17 00:17:41.009 %System %Security AuditChange 740 _SYSTEM
 2025-03-16 00:17:41.950 %System %Security AuditChange 727 _SYSTEM
 2025-03-15 00:28:11.279 %System %Security AuditChange 714 _SYSTEM
 2025-03-14 01:25:11.903 %System %Security AuditChange 702 _SYSTEM
 2025-03-13 00:52:22.262 %System %Security AuditChange 692 _SYSTEM
 2025-03-12 02:33:03.368 %System %Security AuditChange 680 _SYSTEM

2024.3

監査データベースの閲覧

イベントソース	ページサイズ:	結果:	ページ:
*	1000	1000+	[1] の 1
イベントタイプ			
*			
イベント名	*		
SQLステートメントタイプ	DDL		
システムID	*		
PID	*		
ユーザ	* (すべて)		
Admin			
CSPSystem			
IAM			
SuperUser			
UnknownUser			
認証	ハッシュ		
認証なし			
Kerberos			
Kerberos暗号化			
Kerberos/ケケット整合性			
相互TLS承認			
開始日時	<input type="text"/>		
終了日時	<input type="text"/>		

時間

イベントソース イベントタイプ イベント PID CSPセッション ユーザ

2025-04-16 00:43:29.675 %System %Security AuditReport 662 pDKeuzKkpL SuperUser
 2025-04-16 00:43:22.890 %System %Security AuditReport 662 pDKeuzKkpL SuperUser
 2025-04-16 00:01:01.015 %System %Security AuditChange 770 _SYSTEM
 2025-04-16 00:01:01.009 %System %Security AuditChange 770 _SYSTEM
 2025-04-16 00:00:01.007 %System %System ConfigurationChange 767 _SYSTEM
 2025-04-15 21:22:52.079 %System %Login LoginFailure 662 6t4ZmE0A40 UnknownUser
 2025-04-15 17:15:01.528 %System %Login LoginFailure 662 38qH3m1gx1 UnknownUser
 2025-04-15 15:19:35.038 %System %Login LoginFailure 662 pb9u1ZmA4m UnknownUser
 2025-04-15 15:19:14.976 %System %System Start 482
 2025-04-15 15:19:14.975 %System %System Start 482
 2025-04-15 15:19:14.968 %System %Security AuditChange 482
 2025-04-15 04:09:14.121 %System %Security AuditChange 1261 _SYSTEM
 2025-04-15 04:09:14.108 %System %Security AuditChange 1261 _SYSTEM
 2025-04-15 01:51:07.319 %System %System ConfigurationChange 1251 _SYSTEM
 2025-04-14 14:56:37.655 %System %System ConfigurationChange 651 pDKeuzKkpL SuperUser
 2025-04-14 14:56:37.640 %System %System ConfigurationChange 651 pDKeuzKkpL SuperUser
 2025-04-14 14:56:37.613 %System %System ConfigurationChange 651 pDKeuzKkpL SuperUser
 2025-04-14 14:56:37.609 %System %System DatabaseChange 651 pDKeuzKkpL SuperUser
 2025-04-14 14:56:37.608 %System %Security RoleChange 651 pDKeuzKkpL SuperUser

その他の改善点



新しいIRISMETRICSデータベース

IRISの新しいシステムデータベース「IRISMETRICS」が導入されました。これは相互運用性（インターフェラビリティ）の使用状況を保存するためのデータベースです。

- このデータベースは InterSystems IRIS が内部的に使用するためのデータベースです。
- 開発者や利用者が直接アクセスするデータベースではありません。

2024.2

Globals Export Import Find Replace Delete

Look in: Database ✓ ENSLIB

Name	Keep	Collation
DeepSee.Cubes	No	IRIS standard
DeepSee.FolderD	No	IRIS standard
DeepSee.FolderI	No	IRIS standard
DeepSee.FolderItemD	No	IRIS standard
DeepSee.FolderItemI	No	IRIS standard
DeepSee.UserPreferences	No	IRIS standard
Ens.Config.ItemD	No	IRIS standard
Ens.Config.ItemI	No	IRIS standard
Ens.Config.ProductionD	No	IRIS standard
Ens.Config.SearchTablePropC	No	IRIS standard
Ens.Config.SearchTablePropD	No	IRIS standard
Ens.Config.SearchTablePropI	No	IRIS standard
Ens.LegacyPortal	No	IRIS standard
Ens.Rule	No	IRIS standard
Ens.Rule.ActionC	No	IRIS standard
Ens.Rule.Delegates	No	IRIS standard
Ens.Rule.RoutingMap	No	IRIS standard
Ens.Rule.RoutingMapTargets	No	IRIS standard
Ens.Rule.RuleC	No	IRIS standard
Ens.Rule.RuleDefinitionC	No	IRIS standard
Ens.Rule.RuleDefinitionD	No	IRIS standard
Ens.Rule.Targets	No	IRIS standard
Ens.Rule.CacheCounter	No	IRIS standard
EnsEDI.Description	No	IRIS standard
EnsEDI.Schema	No	IRIS standard

2024.3

Globals Export Import Find Replace Delete

Look in: Database ✓ IRISMETRICS

Include: System items

Show:

Filter: Global name *

Maximum rows 1000

Name	Keep	Collation
IRIS.Metrics.InterfaceActID	No	IRIS standard
IRIS.Metrics.InterfaceActI	No	IRIS standard
IRIS.Metrics.InterfaceD	No	IRIS standard
IRIS.Metrics.InterfaceI	No	IRIS standard
IRIS.Msg	Yes	IRIS standard
IRIS.MsgNames	Yes	IRIS standard
ISC.Src.Jrn	Yes	IRIS standard
ISCMethodWhitelist	No	IRIS standard
ROUTINE	Yes	IRIS standard
oddCOM	Yes	IRIS standard
oddDEF	Yes	IRIS standard
oddDEP	Yes	IRIS standard
oddEXT	Yes	IRIS standard
oddEXTR	Yes	IRIS standard
oddMAP	Yes	IRIS standard
oddMETA	Yes	IRIS standard
oddPKG	Yes	IRIS standard
oddPROC	Yes	IRIS standard
oddPROJECT	Yes	IRIS standard
oddPYTHON	Yes	IRIS standard
oddSQL	Yes	IRIS standard



プラットフォームアップデート

AIX上でOpenSSL 1.x サポート終了

AIXでは以下のOpenSSLバージョンがサポートされています：

- OpenSSL 1.0.2 (LTSリリース)
- OpenSSL 1.1.1 (LTSリリース)
- OpenSSL 1.1.2
- OpenSSL 3.0.x (最新は3.0.15.1000)

2025年3月現在、OpenSSL 3.0.15.1000がAIX 7.1、7.2、7.3向けに利用可能となっています

OpenSSL 3.0

- メジャーリリースであり、以前のバージョンからの再コンパイルが必要
- Apacheライセンスv2への変更
- プロバイダー概念の導入とFIPSサポート
- 低レベルAPIの非推奨化

OpenSSL 3.0から低レベルAPI（アルゴリズム固有の関数）は正式に非推奨となり、将来のバージョンでは削除される可能性があるため、すべてのコードを汎用性の高い高レベルAPI（EVPなど）に移行すべきだということです。



2024.2リリースについて

公式ドキュメント:2024.2リリースノート(英文)

https://docs.intersystems.com/iris20242/csp/docbook/DocBook.UI.Page.cls?KEY=GCRN_new20242

開発者向け機能強化

- Windows版からStudio IDEが削除
- 外部テーブル (Foreign Tables) が実験的機能から完全サポートへ移行し
- 新THROUGHコマンドにより外部サーバーへのSQL直接送信が可能に
- Windows環境でPython 3.9以上 (Anaconda含む) の選択が可能に

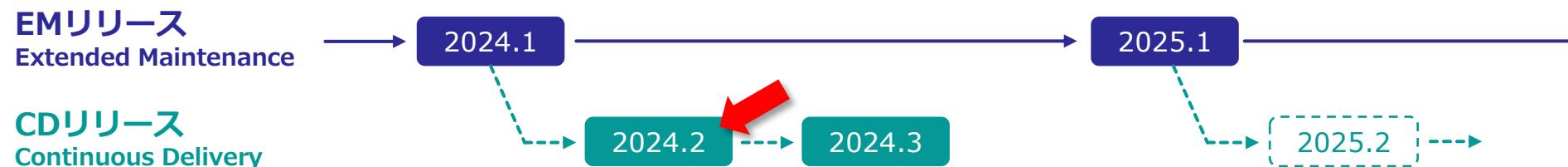
AIおよびアナリティクスの機能強化

- 標準偏差と分散計算用のKPIプラグイン追加
- PowerBIコネクタが「InterSystems Health Insight Connector」に名称変更
- Logi Report 2024.1 SP2によるPDF出力の強化

インフラと性能

- コンテナがUbuntu 24.04ベースに更新
- JDBCドライバ最適化によりデータ取込が最大10倍高速化
- SQL関数(YEAR, ISNULL, NVL, COALESCE)の最適化
- JobServersパラメータが自動的に利用可能なジョブサーバーを維持するよう改善
- Ubuntu 20.04のサポート終了、Ubuntu 24.04のサポート追加

製品のリリースサイクル





開発者向け機能強化

IRIS Studioの削除

2024.2のWindowsインストーラにはIRIS Studioが含まれなくなりました。IRIS Studioを利用する必要がある場合は、[WRC component distribution page](#)からダウンロードし、ご利用ください。

Name	Product	Version	Maint	Build	Os	Arch	File Type	Size (Mb)	Download ...
Web Gateway	IRIS	2024.3	0	217.0	macOS (App...)	64 Bit	tar.gz	2	
Web Gateway	IRIS	2024.3	0	217.0	Windows	64 Bit	exe	33	
Adaptive Analytics	IRIS	2024.1	3	6965.0	Red Hat 7	64 Bit	rpm	1990	
Adaptive Analytics	IRIS	2024.1	3	6965.0	Red Hat 8	64 Bit	rpm	1996	
Adaptive Analytics	IRIS	2024.1	3	6965.0	Ubuntu 20.04	AMD 64	deb	1999	
Adaptive Analytics - UDAF Package	IRIS	2024.1	0	AA-1	All	64 Bit	xml	1	
IRIS Studio	IRIS	2024.1	2	398.0	Windows		msi	16	
iscagent	IRIS	2024.1	3	456.0	AIX	64 Bit	tar.gz	20	
iscagent	IRIS	2024.1	3	456.0	AIX Open SS...	64 Bit	tar.gz	20	
iscagent	IRIS	2024.1	3	456.0	Red Hat 8	64 Bit	tar.gz	22	
iscagent	IRIS	2024.1	3	456.0	Red Hat 9	ARM64	tar.gz	21	
iscagent	IRIS	2024.1	3	456.0	Red Hat 9	64 Bit	tar.gz	21	
iscagent	IRIS	2024.1	3	456.0	Red Hat 8	ARM64	tar.gz	21	
Total Items: 346									

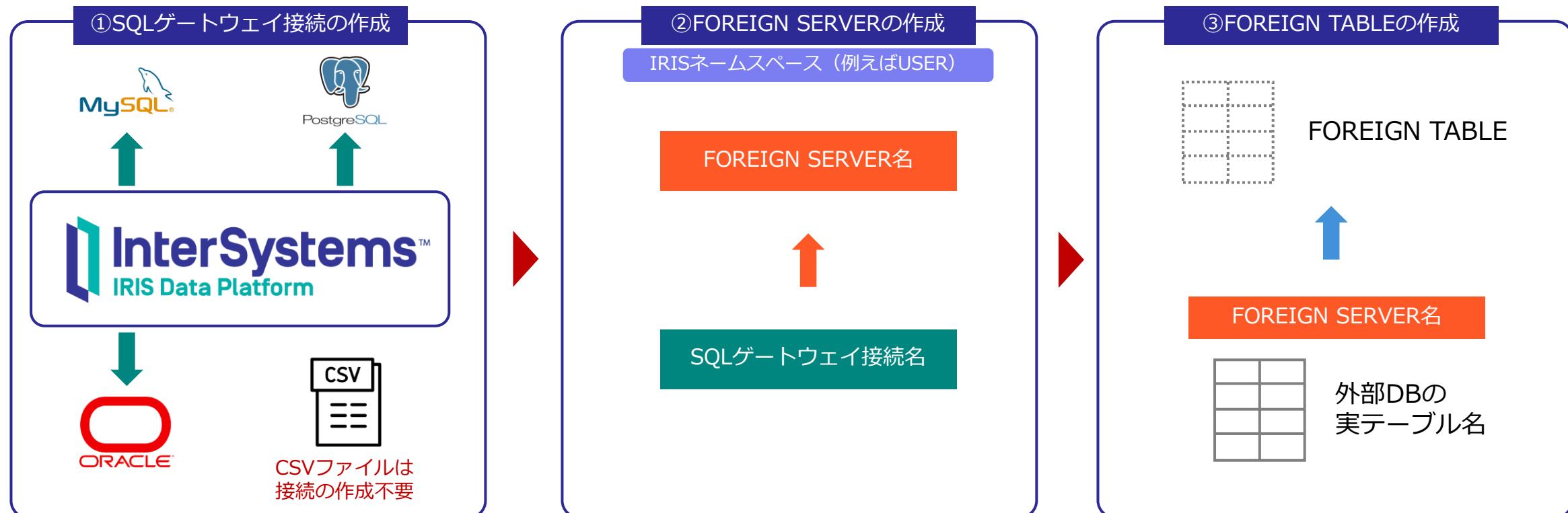
<https://wrc.intersystems.com/wrc/coDistGen.csp>



開発者向け機能強化

外部テーブル(FOREIGN TABLE)のフルサポート

- InterSystemsは2023.1で外部テーブルを実験的機能として導入し、2024.2リリースでは初期ユーザーからのフィードバックに基づき大幅に改善しました。主な強化点はメタデータ管理の向上、述語プッシュダウンの改善、およびLOAD DATAコマンドとの整合性向上です。実験段階から正式版への移行に伴い、ロールモデルも確定しました。
- 新機能としてTHROUGHコマンドが導入され、外部サーバーに直接SQLを送信できるようになりました。これは新しい外部テーブルを作成する前の検証に役立ちます。





開発者向け機能強化

外部テーブル(FOREIGN TABLE)のフルサポート



参考 : FOREIGN TABLEからのSELECT実行例

サーバ iris_20242-foreign ネームスペース USER ユーザ SuperUser ライセンス先 InterSystems IRIS Community インスタンス IRIS
システム > SQL
フィルタ SQLUser.* 選用先:すべて
システム 非推奨
スキーマ
▼ テーブル
> SQLUser.customers
> ビュー
> プロシージャ
> クエリキャッシュ

実行 プラン表示 覆歴を表示 クエリビルダ 表示モード 最大 1000 その他オプション
SELECT COUNT(id) FROM customers
行数: 1 パフォーマンス: 0.2144 秒 1803 グローバル参照 11402 実行されたコマンド 0 ディスク読み込みレイテンシ (ms) クエリ:%sqlcq.USER.cls4 最終更新:2025-04-23 11:42:20.578
Aggregate_1
1
1 行が影響を受けました

ローカルのテーブル(データ)として利用可能

一部DB(MySQL等)を利用したFOREIGN TABLE機能を利用する場合、以下のテキスト識別区切り文字の無効化が必要な場合があります。

SQL: ALTER FOREIGN SERVER FS名 ALTER NODELIMITEDIDS

FS名の箇所は、FOREIGN SERVER名を指定します。
正しく外部テーブルにアクセスできない場合は上記をお試しください。



開発者向け機能強化

Windows環境でEmbedded Python環境のPythonバージョン変更可能

Windowsユーザーも、Embedded Pythonで使用するPythonランタイムを選択できるようになりました。Python 3.9以降（Anacondaを含む）から、組織のニーズに合わせて適切なランタイムを選べます。なお、Linuxユーザーについては、2024.1リリースからすでにこの機能が利用可能になっています。

変更方法：

The screenshot shows the '構成' (Configuration) section of the 'メモリ詳細' (Memory Detail) settings. The path is: ようこそ SuperUser > 構成 > メモリ詳細.

メモリ詳細設定			
以下はシステムのメモリ詳細設定の一覧です:			
設定	値	編集	ヘルプ
BackoffDisabled	偽	編集	ヘルプ
ConsoleFile		編集	ヘルプ
LargePagesDisabled	偽	編集	ヘルプ
LargePagesRequired	偽	編集	ヘルプ
LibPath		編集	ヘルプ
LockSharedMemory	偽	編集	ヘルプ
LockTextSegment	偽	編集	ヘルプ
MaxServerConn	1	編集	ヘルプ
MaxServers	2	編集	ヘルプ
Path		編集	ヘルプ
PythonPath		編集	ヘルプ
PythonRuntimeLibrary		編集	ヘルプ
PythonRuntimeLibraryVersion		編集	ヘルプ
UUIDv1RandomMac	0	編集	ヘルプ
VMSConsoleTerminal		編集	ヘルプ
ZFSSize	0	編集	ヘルプ

2箇所変更する必要があります

- PythonRuntimeLibrary
- PythonRuntimeLibraryVersion

PythonRuntimeLibrary

(Pythonランタイムライブラリのパス)

設定例:

Windows: C:\Program Files\Python311\python3.dll

Linux:

/usr/lib/x86_64-linux-gnu/libpython3.11.so.1.0

PythonRuntimeLibraryVersion

(Pythonのバージョン)

設定例:

3.11

現状のバージョン確認方法

```
USER>do #class(%SYS.Python).Run("import sys; print(sys.version)")  
3.12.3 (main, Sep 11 2024, 14:17:37) [GCC 13.2.0]
```

既定値(未設定)の場合は、Embedded PythonのバージョンはシステムのPythonランタイムバージョンと同じです。



AIおよびアナリティクスの機能強化

• Business Intelligence

BI機能に幾つかの改善がなされます。

- ファクト・テーブルからのメジャーに基づいて標準偏差と分散を計算するための標準 KPI プラグインが追加されました。
- PowerBI コネクタ特権との統合が強化され、PowerBI ユーザがリソースによって制限されていない、またはパブリック・キューブであるすべてのキューブにアクセスできるようになりました。

• Reports

PDFエクスポート機能の改善を含むLogi Report 2024.1 SP2が含まれます。

• Power BI用InterSystemsコネクタの変更

Microsoft Power BI 2024 年 7 月リリースより、インターリームズが Power BI 用に提供するコネクタは、InterSystems IRIS (Beta) Connector から InterSystems Health Insight Connector に名称が変更されました。旧名称のコネクタを使用してレポートまたはダッシュボードを構築したクライアントについては、「既存のレポートまたはダッシュボードのデータ・ソースに接続できない」に記載されている手順に従って、接続を更新してください。



プラットフォームアップデート

コンテナバージョンのIRIS 2024.2のベースOSがUbuntu 24.04に変更

2024.1

システム概要	
バージョン	IRIS for UNIX (Ubuntu Server LTS for ARM64 Containers) 2024.1.2 (Build 398U) Thu Oct 3 2024 14:29:04 EDT
構成	/usr/irissys/iris.cpf
データベースキャッシュ(MB)	4490
ルーチンキャッシュ (MB)	448
ジャーナルファイル	/usr/irissys/mgr/journal/20250416.002
スーパーサーバ・ポート	1972
ウェブサーバポート	52773
ライセンスサーバアドレス/ポート	/
ライセンス先	InterSystems IRIS Community
クラスタサポート	
ミラーリング	
システム開始日時	
暗号化キー識別子	
NLSロケール	
このセッションの優先言語	

```
$ cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.5 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.5 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
```

2024.2

システム概要	
バージョン	IRIS for UNIX (Ubuntu Server LTS for ARM64 Containers) 2024.2 (Build 247U) Tue Jul 16 2024 09:34:42 EDT
構成	/usr/irissys/iris.cpf
データベースキャッシュ(MB)	4490
ルーチンキャッシュ (MB)	448
ジャーナルファイル	/usr/irissys/mgr/journal/20250416.002
スーパーサーバ・ポート	1972
ウェブサーバポート	52773
ライセンスサーバアドレス/ポート	/
ライセンス先	InterSystems IRIS Community
クラスタサポート	
ミラーリング	
システム開始日時	
暗号化キー識別子	
NLSロケール	
このセッションの優先言語	

```
$ cat /etc/os-release
PRETTY_NAME="Ubuntu 24.04.1 LTS"
NAME="Ubuntu"
VERSION_ID="24.04"
VERSION="24.04.1 LTS (Noble Numbat)"
VERSION_CODENAME=noble
ID=ubuntu
ID_LIKE=debian
```



性能/拡張性/セキュリティの強化

SQLのクエリパフォーマンス高速化

- **SQLパフォーマンス**

多数のSQLクエリパフォーマンスの改善

- カラムナー・ストレージを使用して大量のデータをテーブルに取り込む場合、新しいJDBCドライバーの最適化により、取り込みが1桁も速くなります。
- SQL クエリで YEAR、ISNULL、NVL、および COALESCE 関数を使用すると、より効率的なクエリ実行ストラテジにつながる追加の最適化がトリガされるようになりました。
- クエリ処理で、グローバル・テンポラリ・テーブル（複数のワーカ・プロセスで共有）と適応型並列実行（%FIND 演算子の使用時など）が広く使用されるようになりました、全体的なパフォーマンスが向上しました。

- **ジョブサーバの変更**

JobServers パラメータの動作が更新されました。以前は、使用済みのジョブサーバを置き換えるために新しいジョブサーバが作成されることはなく、パラメータは総数と起動時に作成される数の両方を設定していました。

今回の更新により、JobServersパラメータは起動時に作成されるジョブサーバの数を設定しますが、システムが維持する利用可能なジョブサーバの目標数も決定します。ジョブ・サーバーが利用されるについて、より多くのジョブ・サーバーが作成されるため、常に利用可能なジョブ・サーバーが存在します。