IBM Host Access Transformation Services

ユーザーと管理者のガイド

バージョン 9.5

SA88-5379-01

IBM Host Access Transformation Services

ユーザーと管理者のガイド

バージョン 9.5

SA88-5379-01

お願い

本書および本書で紹介する製品をご使用になる前に、 583 ページの『付録 C. 特記事項』に記載されている情報をお読みください。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： SC27-5904-01
IBM Host Access Transformation Services
User's and Administrator's Guide
Version 9.5
Tenth Edition (November 2015)

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

© Copyright IBM Corporation 2003, 2015.

目次

第 1 章 Host Access Transformation Services (HATS) の使用 1

HATS の主要概念とオブジェクト 2
HATS アプリケーション処理の概要 9
HATS アプリケーション開発の理解 11
使用している HATS のバージョンを調べる方法 . . 13
問題判別情報の収集 14
HATS に関する情報の検索場所. 14

第 2 章 HATS V9.5 へのマイグレーション 17

HATS のマイグレーション 17
HATS プロジェクトのインポート 17
HATS マイグレーション・ウィザードの使用 . . 18
HATS 変換のマイグレーション. 19
特別なマイグレーションの考慮事項 19
Host Publisher ユーザーのための HATS 26
Host Publisher バージョン 4 からのマイグレーション. 27
J2EE マイグレーション 30

第 3 章 HATS Web アプリケーションの開発および展開 33

HATS Web アプリケーションの編成 35
HATS Web プロジェクトを別の .ear ファイルに移動させる 36
HATS プロジェクトのバックアップ 37
HATS Web プロジェクトのエクスポートとインポート 37
Web プロジェクトのエクスポート. 37
Web プロジェクトのインポート 38
HATS Web アプリケーションの展開 38
HATS ランタイムの使用可能化. 39
プロジェクトを Java EE アプリケーションとしてエクスポート. 39
ランタイム環境でのアプリケーションのインストール. 40
ランタイム環境で必要な変更 42
Apache Geronimo Server 用の HATS アプリケーションの開発 45
Geronimo サーバー用のインストールおよび構成 45
Geronimo サーバーに関する考慮事項および制約事項. 46
Oracle WebLogic Server 用 HATS アプリケーションの開発 46
WebLogic サーバーのインストールおよび構成. . 46
WebLogic サーバーに関する考慮事項および制約事項. 49
IBM Bluemix Server 用の HATS アプリケーションの開発 49
IBM Bluemix サーバー用の Rational SDP の構成 49
IBM Bluemix Server に関する考慮事項および制約事項. 50
WebSphere Application Server Liberty Profile 用の HATS アプリケーションの開発. 52
WebSphere Application Server Liberty Profile に関する考慮事項および制約事項 53
モバイル装置用の HATS アプリケーションの作成 53
モバイル装置に関する考慮事項および制約事項 . 54
iPad デバイスに関する考慮事項および制約事項 62
Android デバイスに関する考慮事項および制約事項 63

第 4 章 HATS リッチ・クライアント・アプリケーションの開発とデプロイ . . . 65

HATS リッチ・クライアント・アプリケーションの開発. 66
ターゲット・プラットフォームの特性 67
プロジェクト・コンテンツ 67
コンパイラー準拠レベルの設定. 68
HATS RCP ランタイム拡張プロジェクト 69
HATS リッチ・クライアント・プロジェクトの処理 70
Web プロジェクトとリッチ・クライアント・プロジェクトの間でのリソースのコピー 72
HATS リッチ・クライアント・プロジェクトのエクスポートとインポート 73
HATS リッチ・クライアント・アプリケーションのテスト 74
ターゲット・プラットフォームの構成 74
ランタイム・プラグインのインストール. . . . 75
デフォルト JRE の設定 76
プロジェクトの起動 77
HATS リッチ・クライアント・アプリケーションのデプロイ 80
配布用の Eclipse クライアント環境のパッケージ化 80
既存の Eclipse クライアントのパッケージ化 (Eclipse RCP、Lotus Notes、または Lotus Expeditor Client) 81
プラグインおよびフィーチャーのバージョン番号の更新 92
HATS リッチ・クライアント・アプリケーションの管理. 93
ランタイム環境 94
「アプリケーション」ビュー 94
「変換」ビュー 97
ワークステーション ID のプロンプト表示 . . . 99
LU 名のプロンプト表示 99
プリファレンス 99
HATS リッチ・クライアントの考慮事項と制限 . . 103

© Copyright IBM Corp. 2003, 2015

第 5 章 HATS プロジェクトの変更 . . 107
概要 107
接続 109
テンプレート 110
レンダリング 111
デフォルト・レンダリング 111
グローバル規則. 116
テキスト置換 120
コンポーネントおよびウィジェット 120
ツールバー RCP のみ 121
アプリケーション・キーパッド 121
ホスト・キーパッド 122
オペレーター情報域 122
DBCS 124
イベント 124
画面イベントの優先順位. 124
アプリケーション・イベント 125
その他. 129
キーボード・サポート 129
クライアント・ロケール. 130
接続パラメーターのオーバーライド 130
非同期更新 RCP のみ 134
自動切断および最新表示 Web のみ 134
グローバル変数のオーバーライド. 141
クライアント設定 142
マクロ・コンテンツ・アシスタンス 147
ポートレットの設定 148
ソース. 149
HATS プリファレンスの使用 149
その他の Rational SDP プリファレンスの使用 . . 155

第 6 章 接続の管理 157
接続の作成 157
接続エディター. 158
概要 158
基本 158
拡張 159
印刷 162
画面処理 165
セキュリティー. 166
プール. 168
マクロ. 169
ユーザー・リスト 171
ソース. 172

第 7 章 画面イベントの処理 173
「画面カスタマイズの作成」ウィザード 173
「画面組み合わせを作成」ウィザード 174
画面イベントの編集 175
概要 175
画面認識基準または開始画面 175
レンダリング (画面組み合わせのみ) 179
ナビゲーション (画面組み合わせのみ) 180
終了画面 (画面組み合わせのみ) 180
アクション 181
グローバル規則. 194
テキスト置換 195
次の画面 197
ソース. 197
使用禁止画面 198
認識基準 198
使用禁止画面の自動処理. 198
複数の使用禁止画面の処理 199
BMS マップ・セットのインポート 200

第 8 章 変換の操作 203
「変換の作成」ウィザード 203
変換の編集 205
Web プロジェクトの変換の編集 205
リッチ・クライアント・プロジェクトの変換の編集 209
Web 変換とリッチ・クライアント変換の編集の違いの要約 212
「変換」ウィザード 213
ホスト・コンポーネントを挿入 213
ホスト・コンポーネントを編集Web のみ . . . 215
Dojo 編集用に変換 Web のみ 215
デフォルト・レンダリングを挿入. 215
デフォルト・レンダリングを編集Web のみ . . 216
タブ付きフォルダーを挿入Web のみ. 216
マクロ・キーを挿入 217
グローバル変数を挿入 218
オペレーター情報域を挿入Web のみ. 218
ホスト・キーパッドを挿入 218
アプリケーション・キーパッドの挿入 219
すべてのホスト・コンポーネントを挿入Web のみ 220
格納済み画面を挿入Web のみ 220
変換のプレビュー 220
ホスト・キーパッド 221

第 9 章 コンポーネントおよびウィジェットの説明と設定 223
コンポーネントおよびウィジェットの設定. . . . 223
ホスト・コンポーネントの設定 223
コマンド行 224
ダイアログ 224
ENPTUI 228
フィールド 230
ファンクション・キー 231
HTML DDS キーワード Web のみ 233
入力フィールド. 234
ヒント付き入力フィールド 236
項目の選択 238
ライト・ペン (アテンション) 240
ライト・ペン (選択) 241
選択リスト 242
サブファイル 244
テーブル 252
テーブル (フィールド) 254
テーブル (可視) 255
テキスト 257

URL 257
ウィジェット設定 258
ボタン 258
ボタン・テーブル 260
カレンダー Web のみ 262
チェック・ボックス 269
コンボ RCP のみ 271
ダイアログ 274
ドロップダウン (データ入力) 276
ドロップダウン (選択) 279
フィールド 281
グラフ (水平棒、折れ線、垂直棒) 288
ラベル 292
リンク 293
リンク (項目選択) 295
リスト 297
ポップアップ 300
ラジオ・ボタン (データ入力) 305
ラジオ・ボタン (項目選択) 308
ラジオ・ボタン (選択) 311
スクロール・バー (ENPTUI) 313
サブファイル (チェック・ボックス) 314
サブファイル (ドロップダウン) 322
サブファイル (ポップアップ) 329
テーブル 337
テキスト入力 343
ツールバー RCP のみ 347
ウィジェット設定 (Dojo) 351
HATS Dojo ウィジェットの考慮事項と制限 . . 351
コンボ・ボックス (Dojo) Web のみ 352
日付テキスト・ボックス (Dojo) Web のみ . . . 354
拡張グリッド (Dojo) Web のみ 357
フィルター操作選択 (Dojo) Web のみ 359
テキスト・ボックス (Dojo) Web のみ 361
検証テキスト・ボックス (Dojo) Web のみ . . . 362
コンポーネントおよびウィジェットのマッピング 364

第 10 章 テンプレートの使用 367
テンプレートの例 368
「テンプレートの作成」ウィザード 371
テンプレートの編集 372
Web プロジェクトのテンプレートの編集 . . . 372
リッチ・クライアント・プロジェクトのテンプレートの編集 377
アプリケーション・キーパッド 379

第 11 章 マクロとホスト端末 381
マクロ・エディター 382
概要 382
プロンプトおよび抽出 394
ソース 394
マクロ・エラーの処理 395
マクロのインポート 395
マクロのエクスポート 396
マクロのヒント 396
無限ループの回避 396
一時画面の処理 396

第 12 章 グローバル変数との対話 . . . 399
グローバル変数の名前変更 401
グローバル変数とマクロ変数の違い 401
グローバル変数 401
マクロ変数 402

第 13 章 統合オブジェクトの使用 . . . 403
統合オブジェクトの作成 404
統合オブジェクトのチェーニング 405
どのようなときに統合オブジェクト・チェーニングを使用するかの判断 405
統合オブジェクト・チェーニングの使用 . . . 406
統合オブジェクト・チェーニングを使用するアプリケーションのデバッグ 408
統合オブジェクトからの Web ページの構築 . . . 409
モデル 1 Web ページの作成 409
Struts Web ページの作成 410
JSF Web ページの作成 411
BasicIOErrorPage.jsp および AdvancedIOErrorPage.jsp 412
JSP での統合オブジェクトの使用 413
統合オブジェクト・プロパティーの挿入 . . . 413
HATS アプリケーションへの転送の挿入 . . . 414

第 14 章 画面の結合 415
連続出力データの結合 415
不連続出力データの結合 415
複数アプリケーションからの出力の結合 416
複数画面の入力の結合 416

第 15 章 印刷サポートの使用可能化 417
3270 ホストでのホスト印刷セッションの構成 . . 417
プロジェクト用印刷サポートの定義 417
3270E 接続の場合 417
5250 接続の場合 420
ユーザーへの資料の提供 421

第 16 章 キーボード・サポートの使用可能化 423
キーボード・サポートの定義 423
キーパッドの外観の変更 423
ユーザーへの資料の提供 424
特殊な 5250 フィールド・キー・サポート . . . 426
HATS アプリケーションでのキーボードの再マップ 428
HATS キーを再マップする前に理解しておくべき概念 428
HATS Web アプリケーションのキーの再マップ 431
HATS リッチ・クライアント・アプリケーションのキーの再マップ 434
変換への追加のキーパッド・ボタンの追加 . . . 436

第 17 章 ホスト・シミュレーションの使用 437
「ホスト・シミュレーション」ウィザード 437
ホスト・シミュレーション・エディター 438

「概要」タブ 438
「ソース」タブ. 439
実行時環境での記録 439
再生オプション. 439
トレース・ファイルのインポートおよびエクスポート 440

第 18 章 HATS 管理コンソールの使用 443
HATS 管理コンソールおよび WebSphere セキュリティー. 444
HATS 管理コンソールの役割 446
HATS 管理コンソールの開始 447
HATS Toolkit での管理コンソールの開始 . . . 447
HATS 管理コンソールの機能の使用 448
管理の有効範囲の選択 448
ライセンスの管理 449
接続のモニター. 449
接続プールのモニター 449
プール定義のモニター 450
ユーザー・リストとユーザー・リスト・メンバーのモニター 450
問題判別コンポーネントの管理 450
端末の表示機能. 454
テストとデバッグでの端末の表示の使用 . . . 454

第 19 章 WebSphere Portal と HATS 457
HATS ポートレットの作成 457
新規 HATS ポートレット・プロジェクトの作成 458
HATS Web プロジェクトからのポートレットの生成 458
HATS ポートレットの処理 460
接続パラメーターおよびグローバル変数のオーバーライド 460
ポートレット通信 462
カスタム PDT およびカスタム・テーブル. . . 465
Web Services for Remote Portlets 465
HATS ポートレットのテスト 466
HATS ポートレットのエクスポート 466
HATS ポートレットの管理 467
独立型の HATS 管理アプリケーションの使用 467
HATS ポートレットの考慮事項と制限 468
一般的な考慮事項と制限. 468

第 20 章 WebFacing と HATS 473
WebFacing との HATS 対話 473
WebFacing サーバーへの HATS 接続 473
WebFacing アプリケーションとの HATS 相互運用性 474

第 21 章 セキュリティーおよび Web 高速ログオン. 477
SSL セキュリティーの使用可能化 477
SSH セキュリティーの使用可能化 480
Web 高速ログオン (WEL) の使用 481
概要 481
アーキテクチャー 481
インプリメンテーションの計画 483
インプリメンテーション. 483
WEL ログオン・マクロの作成方法 484
ネットワーク・セキュリティー・プラグイン . . 485
Credential Mapper プラグイン 485
初期化パラメーター 487
SSL 鍵ストア・ファイルの作成 (DCAS のみ) 494
Kerberos サービス・チケットの使用 496
Java 2 セキュリティー 497
ポリシー・ファイル 497
HATS アプリケーションに対する IBM Certificate Management の使用 498
鍵データベース・ファイルの作成. 498
認証局からの証明書の要求および保管 499
自己署名証明書の使用 502
証明書の交換 502
証明書管理ツールの考慮事項 503

第 22 章 言語サポート 505
言語コード 506
コード・ページ. 506
エンコードの設定 509
5250 Unicode サポート 509
コード・ページ 937 用のアクセント付き文字の使用 509
コード・ページ 1388 (GB18030) の使用 510
コード・ページ 1390 および 1399 用のホスト・コード・マッピング 510
JIS2004 サポート 511
キーボードおよびディスプレイの文字の再マップ 514
文字置換の優先順位 515

第 23 章 双方向アプリケーション・サポート 517
ソフトウェア環境 518
ホスト端末の操作 518
ホスト端末に関する制限. 520
画面の取り込み. 520
双方向ホスト・コンポーネントの認識 520
ウィジェット方向の制御. 521
ウィジェットの位置合わせの制御. 522
カスタマイズされたコンポーネントとウィジェットの双方向サポート 523
グローバル変数. 523
テキスト置換 523
グローバル規則. 524
ユーザーによる画面方向反転の可能化 525
VT 双方向表示オプション 526
マクロのプロンプトおよび抽出の双方向オプション 526
マクロ SQL 照会の双方向オプション 527
右から左の印刷サポート. 528
BMS マップのサポート 528
統合オブジェクトのサポート 529
ボトムアップ Web サービスのサポート 530
RESTful Web サービスのサポート 531
ポートレット・サポート. 531

ユーザー向けの情報 531
フィールド自動反転 532
フィールド反転. 533
クリップボードの機能 533
オペレーター情報域 (OIA) 533
キーボード・レイヤーの自動切り替え 534
IBM i 5250 Unicode サポート. 534
アラビア語コード・ページの機能. 534
HATS Toolkit の画面方向 534
JSP ページの方向をカスタマイズ. 535
成形の方向 535
アラビア語の選択的形状. 535
対称スワッピングおよび数字スワッピング. . . 536
アラビア (西洋) 数字の入力を無効にする . . . 536
キーボード・レイヤーの自動切り替え 536
ホスト端末に関する制限. 537
画面キャプチャー 537
System i Access for Web でのアラビア語印刷セッションを設定. 537
その他の考慮事項 538
HATS ファイルへの追加 538

第 24 章 2 バイト文字セットのサポート 541
DBCS および SBCS フィールドのサポート . . . 541
データ・タイプの検査 542
フィールド長の検査 542
コピー/貼り付け操作の防止. 542
Input Method Editor (IME) 543
AutoIME 切り替え. 543
シフトアウト/シフトインに関する考慮事項 . . . 544
その他の考慮事項 544
プロジェクト・テーマ設定 544
プロジェクト設定エディター 544
画面イベント・エディター 547
ホスト・コンポーネント. 547
ウィジェット 548
マクロ・サポート 549
統合オブジェクトの作成. 549
モバイル装置アプリケーションの操作 549
キーボードおよびディスプレイの文字の再マップ 550
ユーザー定義文字の処理. 550

付録 A. ランタイム・プロパティー・ファイル 557
トレース・オプション 561
Host On-Demand トレース 563
ホスト・シミュレーション・トレース 565

付録 B. HATS 画面確定のリファレンス 567
画面確定の概要. 567
画面確定手順 567
アウトバウンド・データの分析 568
OIA フラグの取得待ち 571
カスタマイズ設定の変更. 571
HATS で使用されている計画の判断 572
コンテンション解消 (TN3270E のみ) 573
z/OS Communications Server を使用する場合のコンテンション解消 573
他の Communications Server を使用する場合のコンテンション解消 574
コンテンション解消を使用する場合のパフォーマンスへの影響 574
コンテンション解消状況の判断 574
自動最新表示 576
一時画面の処理. 576
初期ブランク画面処理設定 577
関連する HATS 設定. 578
HATS 画面確定の調整 580
リファレンス 582
コンテンション解消の情報 582
拡張マクロ・ガイド 582

付録 C. 特記事項 583
プログラミング・インターフェース情報 584
商標 585

用語集 587

索引 597

第 1 章 Host Access Transformation Services (HATS) の使用

IBM® Host Access Transformation Services (HATS) では、IBM z Systems プラットフォームで実行される 3270 アプリケーション用、および IBM i オペレーティング・システム・プラットフォームで実行される 5250 アプリケーション用の使いやすいグラフィカル・ユーザー・インターフェース (GUI) を提供する Web アプリケーションやリッチ・クライアント・アプリケーションを作成することができます。HATS アプリケーションでは、オンライン・トランザクション処理 (OLTP) 能力を必要とすることなく 5250 アプリケーションにアクセスできます。本書では、これらの文字ベースの 3270 および 5250 アプリケーションをホスト・アプリケーションと呼びます。また、ホスト・アプリケーション内のビジネス・ロジックとトランザクションへの標準プログラミング・インターフェースを提供する Web サービスを使用して、サービス指向アーキテクチャー (SOA) 資産も作成できます。ビデオ端末 (VT) エミュレーション画面のデータにもアクセスできます。

HATS アプリケーションの外観を最新のものにすることができます。HATS Web アプリケーションを開発するときに、お客様の企業の Web ページまたはポータル・ページに対応したインターフェースを使用できます。また、ユーザーは各自の Web ブラウザーからこの Web アプリケーションにアクセスできます。また、携帯電話、データ収集端末、および携帯情報端末 (PDA) などのモバイル装置からご使用のホスト・アプリケーションへのアクセスを提供する HATS Web アプリケーションを開発することもできます。

ユーザーのデスクトップをターゲットとしたネイティブ・クライアント・アプリケーションを実現するため、Eclipse リッチ・クライアント・プラットフォーム (RCP) インプリメンテーション、Lotus Notes®、または Lotus® Expeditor Client で実行する HATS リッチ・クライアント・アプリケーションを開発できます。Eclipse RCP 環境の詳細については、http://www.eclipse.org/home/categories/rcp.php を参照してください。 Lotus Notes の詳細については、Lotus Notes Knowledge Center (http://www-01.ibm.com/support/knowledgecenter/SSKTMJ_8.5.3/welcome_Domino_8_5_3.html) を参照してください。 Lotus Expeditor Client の詳細については、Lotus Expeditor Knowledge Center (http://www-01.ibm.com/support/knowledgecenter/SSVHEW_6.2.0/welcome.html) を参照してください。

HATS V9.5 の新機能については、*「HATS スタートアップ・ガイド」*を参照してください。

注: 本書では、特に明記されていない限り、HATS Web アプリケーションおよび EJB アプリケーションにのみ適用される構成設定に Web のみ というマークが付いています。HATS リッチ・クライアント・プラットフォーム・アプリケーションにのみ適用される構成設定には、RCP のみ というマークが付いています。

HATS には次の 2 つのコンポーネントがあります。

© Copyright IBM Corp. 2003, 2015

- HATS Toolkit は、Eclipse ベースの IBM Rational® Software Delivery Platform (Rational SDP) 向けに作られたプラグインのセットです。サポートされている Rational SDP のバージョンの詳細については、「*HATS スタートアップ・ガイド*」のセクション『前提条件』を参照してください。

  HATS Toolkit を使用すると、変更内容をその都度プレビューし、保存しながら、新規アプリケーションを一度に 1 ステップずつ作成できます。作成した HATS アプリケーションは必要に応じて簡素化し、データの表示をホスト・アプリケーションよりも使いやすくしたり、ホスト・アプリケーションから HATS アプリケーションへ機能を移動したりできます。HATS Web アプリケーションおよびリッチ・クライアント・アプリケーションの構築開発過程は類似しています。詳しくは、11 ページの『HATS アプリケーション開発の理解』を参照してください。

  HATS アプリケーションの開発後に、このアプリケーションを実動ランタイム環境に展開します。
- HATS ランタイム・コードは、実動ランタイム環境、WebSphere® Application Server、WebSphere Portal、Apache Geronimo Web アプリケーション・サーバー (Tomcat Web コンテナー付属)、Oracle WebLogic Server、またはリッチ・クライアント・プラットフォームにデプロイされた HATS アプリケーションの一部として実行されます。ユーザーは HATS GUI を介して HATS アプリケーションを対話型で操作し、ユーザーとホスト・アプリケーションの間でデータが送受信されます。

詳しくは、33 ページの『第 3 章 HATS Web アプリケーションの開発および展開』および 65 ページの『第 4 章 HATS リッチ・クライアント・アプリケーションの開発とデプロイ』を参照してください。

HATS の主要概念とオブジェクト

このセクションでは、HATS の主要な概念とオブジェクトについて説明します。一部の概念については、9 ページの『HATS アプリケーション処理の概要』で説明します。HATS の主要オブジェクトの多くはウィザードを使用して作成されますが、作成後に表示または修正する場合はエディターを使用します。

プロジェクト

HATS リソース (成果物とも呼ばれる) の集合。これは HATS Toolkit のウィザードを使用して作成され、HATS Toolkit エディターを使用してカスタマイズされます。これらのリソースは、HATS アプリケーションとしてエクスポートされます。HATS プロジェクトのタイプには、Web、EJB、リッチ・クライアントの各プロジェクトと、HATS Web (EJB を含む) アプリケーションの管理を目的とする HATS 管理コンソール・プロジェクトがあります。 詳しくは、33 ページの『第 3 章 HATS Web アプリケーションの開発および展開』を参照してください。

イベント

到達した特定の状態に基づいて一連のアクションを実行する HATS リソース。HATS イベントには、アプリケーション・イベントと画面イベントの 2 種類があります。

アプリケーション・イベント

HAT イベントは、アプリケーションのライフ・サイクル内の状態変更によっ

て起動されます。アプリケーション・イベントとは、例えば、ユーザーが初めて HATS アプリケーションにアクセスする (「開始」イベント)、アプリケーションが認識しない画面を検出する (「一致しない画面」イベント)、などです。詳しくは、125 ページの『アプリケーション・イベント』を参照してください。

画面イベント
特定の画面認識基準に一致することでホスト画面が認識された時点で起動される HATS イベントです。画面イベントには、画面カスタマイズと画面組み合わせの 2 種類があります。

画面カスタマイズ
ホスト画面が認識された時点で一連のアクションを実行するように設計された HATS 画面イベントです。画面カスタマイズの例としては、画面の認識、ユーザーの GUI への画面の変換、マクロの再生による画面のスキップなどがあります。画面カスタマイズ定義には、画面認識基準のセットと、ホスト画面が画面認識基準に一致した場合に実行するアクションのリストが含まれています。画面レベルのグローバル規則とテキスト置換設定も含まれます。詳しくは、173 ページの『第 7 章 画面イベントの処理』を参照してください。

画面組み合わせ
連続する類似ホスト画面から出力データを収集して組み合わせ、1 つの出力ページに表示するように設計された HATS 画面イベントです。画面組み合わせの例としては、部分データのみが含まれている画面を認識し、すべての後続画面をナビゲートして、ユーザーに対して表示するために残りのデータをすべて収集することがあります。画面組み合わせ定義には、組み合わせる開始画面と終了画面の画面認識基準のセット、画面間のナビゲート方法、および各画面から収集したデータを認識してレンダリングするために使用するコンポーネントとウィジェットが含まれています。また、1 回以上実行されるアクションのリスト、画面レベルのグローバル規則とテキスト置換設定も含まれています。詳しくは、173 ページの『第 7 章 画面イベントの処理』を参照してください。

画面認識基準
画面イベントの作成時に、HATS が 1 つ以上の画面の突き合わせに使用する画面認識基準を設定します。画面上のフィールド数または入力フィールド数、カーソル位置の座標、および、画面上に定義された長方形枠内または画面の任意の場所にあるテキスト・ストリングなどの基準を任意に組み合わせて、ホスト画面を認識させることができます。グローバル変数、色、値と比較する領域、画面認識基準の領域基準と比較する領域を使用することもできます。

ホストがなんらかの画面を表示すると、HATS は、そのホスト画面が、プロジェクト内の画面イベントで設定されている画面認識基準のいずれかに一致するかどうかを調べます。HATS が一致を検出すると、画面イベントで定義されているアクションが実行されます。

画面認識基準の設定方法の詳細については、175 ページの『画面認識基準または開始画面』を参照してください。

アクション
イベントが発生した場合 (画面イベント用に指定された画面認識基準にホスト画面が一致した場合など) に実行されるステップ。アクションのリストは、各アプリケーション・イベントおよび画面イベントの定義の一部です。

変換

ホスト・コンポーネントの抽出方法と GUI でのウィジェットを使用した表示方法を定義する JavaServer Pages (JSP) ファイル (Web プロジェクト) または Standard Widgets Toolkit (SWT) コンポジット (リッチ・クライアント・プロジェクト)。変換の適用は、画面イベントの考えられるアクションの 1 つです。パレットから HATS コンポーネントをドラッグし、HATS「プロパティー」ビューから設定を変更して、変換を WYSIWYG 方式で編集することができます。 Java™ ビジュアル・エディターは Rational SDP V9.5 で使用できなくなったため、リッチ・クライアント変換を WYSIWYG 形式で編集することはできません。Rational Application Developer で削除された機能や非推奨になった機能の詳細については、http://www.ibm.com/support/docview.wss?uid=swg27038549 を参照してください。

変換の作成方法の詳細については、 203 ページの『第 8 章 変換の操作』を参照してください。

ホスト・コンポーネント

ホスト・コンポーネントは、HATS アプリケーションのユーザーに対して表示するホスト画面のエレメント (コマンド行、ファンクション・キーなど) を認識する HATS オブジェクトです。HATS が提供するホスト・コンポーネント・セットを使用することも、ユーザー独自のホスト・コンポーネントを作成することもできます。

HATS プロジェクトで使用するホスト・コンポーネントを選択する方法の詳細については、 213 ページの『ホスト・コンポーネントを挿入』を参照してください。

ウィザードを使用してカスタム・コンポーネントおよびウィジェットを作成する方法の詳細については、ご使用のアプリケーション環境に応じて、「*HATS Web アプリケーション・プログラマーズ・ガイド*」または「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

ウィジェット

ウィジェットは、HATS プレゼンテーションにホスト・コンポーネントの GUI 出力を作成する HATS オブジェクトです。例えば、ファンクション・キー・ホスト・コンポーネントをボタン・ウィジェットに変換することで、ファンクション・キーを HATS アプリケーションのプッシュボタンとしてユーザーに対して表示できます。HATS が提供するウィジェットを使用することも、ユーザー独自のウィジェットを作成することもできます。

HATS プロジェクトで使用するウィジェットを選択する方法の詳細については、 213 ページの『ホスト・コンポーネントを挿入』を参照してください。

ウィザードを使用してカスタム・コンポーネントおよびウィジェットを作成する方法の詳細については、ご使用のアプリケーション環境に応じて、「*HATS Web アプリケーション・プログラマーズ・ガイド*」または「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

接続

ホスト・アプリケーションへ接続するために HATS が使用するパラメーターのセット。HATS には、デフォルト接続 (変換接続とも呼ばれる) とバックグラウンド接続の 2 つの接続のタイプがあります。各 HATS アプリケーションでは、HATS によって変換される画面を持つホスト・アプリケーション用のデフォルト接続を 1 つ保持しています。バックグラウンド接続は、HATS アプリケーションにおけるデフォルト接続以外のすべての接続です。HATS では、バックグラウンド接続からの画面の変換は実行されません。ただし、デフォルト接続として使用する接続を動的に選択できます。詳しくは、 157 ページの『第 6 章 接続の管理』を参照してください。

グローバル規則

変換フラグメントを使用して、画面レベルまたはプロジェクト・レベルにおける特定タイプのホスト入力フィールドから特定のウィジェットへの置換を指定する規則。変換フラグメントには、特定の変換においてあるパターンの全オカレンスを置換する際に使用する内容が含まれています。

注: 画面イベントの変換アクションを編集することによって、変換の特定の使用法に対してグローバル規則をオフにすることができます。

詳しくは、 111 ページの『レンダリング』を参照してください。

デフォルト・レンダリング

特定の画面イベントが指定されていないホスト画面をレンダリングするときに HATS が使用するレンダリング方法。ホスト画面の定義領域に適用される特定の変換において、デフォルト・レンダリングを使用できます。 詳しくは、 111 ページの『レンダリング』を参照してください。

テキスト置換

テキスト置換は、GUI 変換において、プロジェクト・レベルまたは画面レベルでホスト画面のテキストをイメージ、HTML (Web のみ)、またはその他のテキストに変換するために使用されます。詳しくは、 120 ページの『テキスト置換』を参照してください。

テンプレート

プロジェクトの外観を拡張可能にする JSP ファイル (Web プロジェクト) または Java SWT コンポジット・クラス (リッチ・クライアント・プロジェクト)。HATS プロジェクトを作成するときに、プロジェクトのデフォルト・テンプレートとして使用するテンプレートを選択します。テンプレートには、会社のロゴや情報、およびほかのページへのリンクを含めることができます。HATS が提供するサンプル・テンプレートから選択してデフォルトとすることも、 HATS Toolkit でテンプレート・ウィザードとエディターを使用してプロジェクト用のカスタム・テンプレートを設計することもできます。既存ファイルから事前設定されたテンプレート、または Web プロジェクトの場合は URL から事前設定されたテンプレートも作成できます。

GUI では、変換が適用されたエリアをテンプレートが囲みます。画面イベントの結果として、デフォルト・テンプレート以外のテンプレートを適用することもできます。テンプレートと変換を関連付ける方法の詳細については、 367 ページの『第 10 章 テンプレートの使用』を参照してください。

ビジネス・ロジック

イベント (画面カスタマイズなど) 内でアクションとして起動される Java コード。ビジネス・ロジックは、アプリケーションに固有です。

ビジネス・ロジックの詳細については、ご使用のアプリケーション環境に応じて、「*HATS Web アプリケーション・プログラマーズ・ガイド*」または「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

グローバル変数

HATS アプリケーション・インスタンスの存続期間中を通して使用できる値を格納するために使用する変数。グローバル変数の値は、ホスト画面から抽出することも、開発者が定義することもできます。グローバル変数は、テンプレート、変換、画面イベント・アクション、または認識基準で使用できます。これらの変数は、エンタープライズ・アーカイブ (.ear) ファイルにおける Web アプリケーション間、または同じリッチ・クライアント環境内で稼働するリッチ・クライアント・アプリケーション間で共用できます。グローバル変数は、索引を付けたり、マクロ、ビジネス・ロジック、統合オブジェクト内で使用したりできます。

グローバル変数の詳細については、 399 ページの『第 12 章 グローバル変数との対話』を参照してください。

マクロ

一組の画面と、それらの画面上で実行するアクションを定義する XML スクリプト。マクロは、ユーザーとホストとの対話を自動化するために使用します。画面をスキップし、ループし、ユーザーにデータ入力を求める (または、自動的にデータを挿入したり、グローバル変数を指定したりする) プロンプトを表示し、ホスト画面情報を抽出する、という動作をマクロに記録して実行できます。

HATS アプリケーションでは、さまざまな方法でマクロを使用できます。

- マクロは、HATS イベントの「**マクロを実行**」アクションの一部としてデフォルト接続で実行できます。マクロは、画面イベントで定義する最後のアクションにする必要があります。
- マクロは、HATS イベントの「**マクロ・トランザクションを実行**」アクションの一部としてバックグラウンド接続で実行できます。
- 変換に「マクロ」ボタンを追加すると、マクロが実行できます。これにより、ユーザーがマクロを実行するかどうかを制御できるようになります。
- 統合オブジェクトによってマクロを実行できます。
- 接続マクロを使用して、接続の事前準備ができます。
- 切断マクロを使用して、接続をクリーンアップできます。

HATS 環境にマクロを取り込む方法の詳細については、 381 ページの『第 11 章 マクロとホスト端末』を参照してください。

統合オブジェクト

マクロから作成される Java オブジェクト。このオブジェクトは、WebSphere アプリケーションのビルディング・ブロックとして使用できます。統合オブジェクトは、ホスト・データ・ソースとの対話をカプセル化する Java Bean で

す。これらのデータ・ソースには、3270、5250、およびビデオ端末 (VT) データを使用する端末向けアプリケーションが含まれています。

同じ接続を共用する統合オブジェクトをグループ化し、HATS アプリケーション内で単一のメジャー・タスクを実行することができます。これは、統合オブジェクト・チェーニングといいます。詳しくは、403 ページの『第 13 章 統合オブジェクトの使用』を参照してください。

ホスト端末

HATS Toolkit でのライブ・ホストへの接続の 1 つ。ホスト端末を使用して、画面の取り込み、画面イベントおよび画面変換の作成、そしてマクロの記録および編集を行うことができます。すでに記録またはインポートしてあるマクロを実行することもできます。ホスト端末では、画面を GUI としてプレビューすることもできます。

端末の表示

テストおよびデバッグ時に、実行時の HATS アプリケーションとホスト・アプリケーションとの対話を監視する目的で使用できるホスト画面を表示する端末ウィンドウ。端末ウィンドウのホスト画面を使用してホスト・アプリケーションと対話することもできます。端末の表示の詳細については、454 ページの『テストとデバッグでの端末の表示の使用』を参照してください。

画面キャプチャー

ホスト画面の XML 表現。.hsc ファイルに保管され、画面カスタマイズ、画面組み合わせ、変換、グローバル規則、またはマクロの作成またはカスタマイズに使用される。画面キャプチャーは、ホストに接続していない場合でも HATS プロジェクトの開発に使用できるため、有用である。 HATS 統合オブジェクトと Web サービス・サポートのコアであるマクロの作成にも有用である。

ビデオ端末 (VT) ホスト画面の画面キャプチャーは、Visual Macro Editor を使用して、およびプール構成時のチェックイン画面として、マクロを作成またはカスタマイズするために使用できる。これらの画面キャプチャーは、画面カスタマイズ、画面組み合わせ、変換、デフォルト・レンダリング、またはグローバル規則の作成には使用できない。

サーバーで実行 Web のみ

Rational SDP の機能の 1 つ。これを使用すると、HATS Web プロジェクトを、WebSphere Application Server インスタンスまたは WebSphere Portal インスタンスのいずれか適切なインスタンスでテストできます。このモードでは、ランタイム環境に展開されたランタイム設定 (runtime.properties ファイルにおいて定義) を変更およびテストできます。このモードでテスト中に変更したランタイム設定は保持されます。この設定は、HATS アプリケーションをランタイム環境に展開するときに有効になります。ランタイム設定の変更の詳細については、450 ページの『問題判別コンポーネントの管理』を参照してください。

サーバーでデバッグ Web のみ

「サーバーで実行」と同じですが、さらに以下の操作が可能です。

- プロジェクトのテスト中に、ディスプレイ端末を使用して、ホスト画面をナビゲートしながら表示する。
- Rational SDP コンソールにデバッグ・メッセージを表示する。

- テスト・サーバー上でアプリケーションを再始動せずに、プロジェクトの変更 (テンプレートや変換の変更など) を表示する。
- runtime.properties ファイルに定義された設定を変更せずに、runtime-debug.properties ファイルに定義されたランタイム設定を変更およびテストし、それをランタイム環境に展開する。
- HATS ビジネス・ロジックなどの Java コードをステップスルーする。

サーバーでプロファイル作成 Web のみ

「サーバーで実行」と同じですが、さらに冗長性を排除するため、ほとんどの場合に必要とされる操作を洗い出し、繰り返し実行されるアクションを特定することができます。パフォーマンス分析の目的でこの機能を使用すると、アプリケーションをより適切に理解できます。

実行 RCP のみ

Rational SDP の機能の 1 つ。これを使用すると、Eclipse、Lotus Notes、または Lotus Expeditor Client の各インスタンスで HATS リッチ・クライアント・プロジェクトをテストできます。このモードでは、ランタイム環境に展開されたランタイム設定 (runtime.properties ファイルにおいて定義) を変更およびテストできます。このモードでテスト中に変更したランタイム設定は保持されます。この設定は、HATS アプリケーションをランタイム環境に展開するときに有効になります。ランタイム設定の変更の詳細については、93 ページの『HATS リッチ・クライアント・アプリケーションの管理』を参照してください。

デバッグ RCP のみ

「実行」と同じですが、さらに以下の操作が可能です。

- プロジェクトのテスト中に、ディスプレイ端末を使用して、ホスト画面をナビゲートしながら表示する。
- Rational SDP コンソールにデバッグ・メッセージを表示する。
- アプリケーションを再始動せずに、プロジェクトの変更 (テンプレートや変換の変更など) を表示する。
- runtime.properties ファイルに定義されたランタイム設定を変更せずに、runtime-debug.properties ファイルに定義されたランタイム設定を変更およびテストし、それをランタイム環境に展開する。
- HATS ビジネス・ロジックなどの Java コードをステップスルーする。

プロファイル RCP のみ

「実行」と同じですが、さらに冗長性を排除するため、ほとんどの場合に必要とされる操作を洗い出し、繰り返し実行されるアクションを特定することができます。この機能をパフォーマンス分析に使用して、アプリケーションの理解を向上させるために役立てることができる。

印刷サポート

開発者がホスト・セッションに関連付けるプリンター・セッションを指定したり、ユーザーがホスト・アプリケーション印刷ジョブを表示したり、ジョブのプリンターへの送信またはディスクへの保存を行う機能。印刷サポートは、デフォルト接続でのみ使用できます。

印刷サポートの詳細については、417 ページの『第 15 章 印刷サポートの使用可能化』を参照してください。

キーボード・サポート

GUI でアプリケーションが稼働しているときに、ユーザーが物理的なキーボードを使用してホストと対話できるようにするための開発者の機能。開発者は、ホスト・キーパッドまたはアプリケーション・キーパッド、またはその両方をプロジェクトに組み込むかどうかも決定します。キーパッドを組み込む場合、開発者は、組み込むキーとそのキーおよびキーパッドを GUI で表示する方法を決定します。

キーボード・サポートの詳細については、 423 ページの『第 16 章 キーボード・サポートの使用可能化』を参照してください。

キーパッド・サポート

ホスト・キーパッドとは、ユーザーがキーボード上の物理的なキーを押すのと同じようにホストと対話できるようにするボタンまたはリンクのテーブルです。ただし、ホスト・キーパッドのボタンやリンクを使用せずに、キーボード上の物理的なキーを使用することもできます。ホスト・キーパッドの詳細については、 221 ページの『ホスト・キーパッド』を参照してください。

アプリケーション・キーパッドとは、ユーザーが HATS アプリケーションに関連するタスク (印刷ジョブの表示や画面の最新表示など) を実行できるようにするボタンまたはリンクのテーブルです。アプリケーション・キーパッドの詳細については、 379 ページの『アプリケーション・キーパッド』を参照してください。

HATS アプリケーション処理の概要

HATS プロジェクトを作成する前に、HATS がホスト・アプリケーションをどのように処理するかを理解しておく必要があります。ユーザーがアプリケーションの各画面にアクセスするとき、HATS は以下に示す手順に従ってアプリケーションを処理します。 11 ページの図 1 はこの手順の流れを示しています。これらの主要概念については、 2 ページの『HATS の主要概念とオブジェクト』で説明します。

1. ホストが画面を表示すると、HATS は、そのホスト画面を、このプロジェクトの使用可能な画面イベントに定義されている画面認識基準と比較します。この比較は、イベント優先順位で定義されている順序に従って、一致が検出されるまで続けられます。
2. 一致が検出されると、HATS はその画面イベントで定義されているアクションを実行します。画面カスタマイズ・イベントと画面組み合わせイベントの両方に、以下のアクションを組み込むことができます。画面組み合わせイベントの場合は、さらに、HATS が複数画面をナビゲートし、各画面からユーザーに対して表示するデータを収集します。
   - 関連するテンプレートを使用して変換を適用。HATS は、ホスト・コンポーネント (変換で定義されているもの) を GUI ウィジェットとして表示します。
   - ビジネス・ロジックを実行
   - グローバル変数を抽出
   - データを挿入
   - グローバル変数を設定
   - グローバル変数を除去
   - グローバル変数を送信 (HATS 標準ポートレット・プロジェクトのみ)

- URL または SWT コンポジットを表示 RCP のみ
- URL を表示 Web のみ
- URL に転送 Web のみ
- マクロを実行
- マクロ・トランザクションを実行
- キーを送信
- 切断
- 一時停止

詳しくは、 181 ページの『アクション』を参照してください。

注:

1. 一致画面が検出されると、1 つ以上のアクションが実行される必要がありますが、ユーザーに対して表示するための画面変換は必須ではありません。つまり、「変換を適用」アクションは必須ではありません。
2. ユーザーがホスト画面データを操作する前と操作した後で、ホスト画面データに対してアクションを実行できます。例えば、グローバル変数としていくらかのデータを抽出し、一部の情報を新しいデータで置き換えてから、ユーザーに対して変換を表示する前に変換を適用することができます。画面データを戻すアクションをユーザーが実行した後、データが最終的にホストに戻される前に、追加の抽出と挿入を実行できます。

3. 画面認識基準がホスト画面に一致しない場合、HATS は不一致画面アプリケーション ・イベントを処理します。このイベントに対するデフォルトのアクションでは、デフォルトの変換を使用し、デフォルト・テンプレートを使用して、ホスト画面が表示されます。デフォルトの変換では、プロジェクト設定で定義されたレンダリング設定が使用されます。
4. ホストがアプリケーションの新たな画面を表示するたびに、HATS は、再びステップ 1 からすべての手順を繰り返します。 11 ページの図 1 に、HATS が画面を処理する仕組みを示します。

   注: 画面スキップ処理のためにマクロを使用している場合、スキップされる画面にはこのプロセスは適用されません。

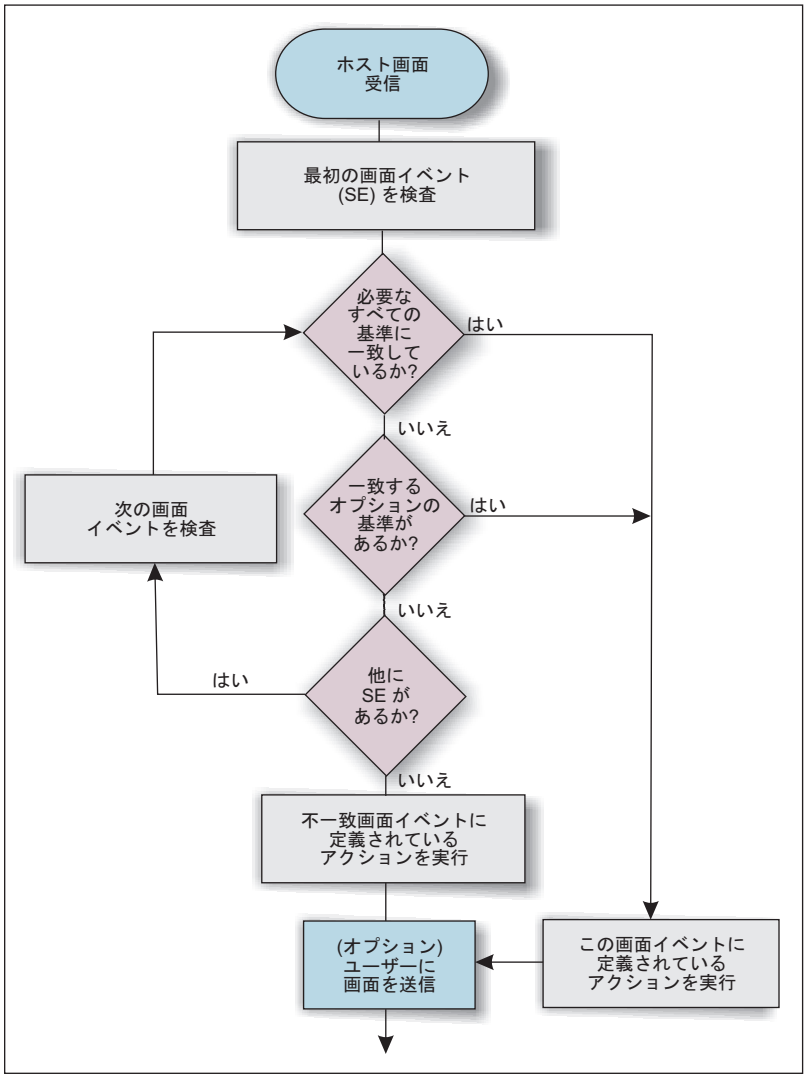


図 1. HATS の画面処理

画面イベントは、HATS プロジェクトの開発における重要な概念です。画面イベントで画面認識基準を定義しておかないと、HATS はホスト画面の受信時に実行するアクションがわかりません。

開発プロセスに着手する前に、画面イベントの基本原則をよく理解しておいてください。画面認識基準のコア・エレメントについては、173 ページの『第 7 章 画面イベントの処理』 で説明します。

HATS アプリケーション開発の理解

HATS を使用すると、アプリケーション開発で反復のアプローチをとることができます。簡単な構成から始め、準備ができたら改良を加えます。Rational SDP のテスト環境では、変更を加えるたびにその変更をテストできます。ほとんどの変更点は、HATS 変換を最新表示するだけで表示できます。必要に応じて修正を行い、作業内容を保存してから次の変更に進むことができます。このセクションでは、HATS アプリケーションを設計および開発する際に使用できる手順の 1 つを説明します。これらの作業は必要な分だけ実行し、いつでもやめることができます。

1. まず、すべてのデフォルトを使用する HATS アプリケーションを作成します。GUI ですべてのホスト画面が即時にレンダリングされます。デフォルト・レンダリングでは、画面の元のレイアウトを保持し、かつユーザーの生産性を向上する GUI コントロール (リンク、ボタン、テーブルなど) が自動的に追加される形でホスト画面がレンダリングされます。
2. 企業の Web ページに対応するテンプレート、またはその他の社内アプリケーションを作成し、企業の外観でレンダリングされるすべてのホスト画面を確認します。HATS に組み込まれているテンプレートの 1 つを使用してこのテンプレートを変更できます。また HATS Web アプリケーションの場合は、企業の Web サイトをテンプレートとしてインポートできます。詳細については、367 ページの『第 10 章 テンプレートの使用』を参照してください。 これで、HATS アプリケーションが企業の Web ページまたはその他の社内アプリケーションのスタイルと色に対応します。
3. プロジェクト・レベル (アプリケーション全体) 設定を構成し、特定の画面を構成せずにホスト・アプリケーションをどのくらい処理できるかを確認します。プロジェクト・レベルの特性は、プロジェクト設定エディターの各種タブで変更できます。詳細については、107 ページの『第 5 章 HATS プロジェクトの変更』を参照してください。例えば、次に示す変更を行います。
   - 選択リストをドロップダウン・リストとしてレンダリングするようにデフォルト・レンダリングを変更します。これにより、すべての選択リストがドロップダウン・リストに変更されます。
   - 1 つ以上のグローバル規則を作成して、HATS でホスト画面上の入力フィールドを変換する方法を設定します。例えば、日付フィールドをカレンダー・ウィジェットに、ロケーション・フィールドをドロップダウン・リストに変更します。
   - テキスト置換を使用して、ホスト画面に表示される 1 つ以上のストリングを変更します。例えば、「**サインオン**」という語を「**ログオン**」に変更できます。
   - ホスト画面の変換方法を確認します。多くの画面に表示される要素で、変換方法を変えたいものがあるかどうかを調べます。それらの要素は、コンポーネント (ホスト・コンポーネントの認識方法を変更する場合) やウィジェット (変換での GUI コントロールのレンダリング方法を変更する場合) の設定変更の候補になります。これらの設定をプロジェクト・レベルで変更して、ホスト画面で使用するデフォルトを設定します。また、コンポーネントやウィジェットを変換に追加するときに、コンポーネントやウィジェットの各インスタンスの設定を変更することもできます。詳しくは、111 ページの『レンダリング』を参照してください。
4. 特定のホスト画面を認識し、実行対象の一連のアクションを起動する画面レベル・イベントを作成します。
   - BMS マップを使用している場合は、最初に BMS マップ・セットをインポートしてホスト画面の画面キャプチャーを作成できます。それ以外の場合は、HATS ホスト端末機能を使用してホスト・アプリケーションに接続し、認識してアクションを実行する特定の画面の画面キャプチャーを作成します。次に各画面キャプチャーから、画面イベントを作成します。
   - いくつかのホスト画面を簡素化し、使用していない項目を除去し、ユーザー入力を簡単にするためにデータを自動的に抽出して取り込むため、「画面カスタ

マイズの作成」ウィザードを使用してこれらの画面をカスタマイズします。ユーザーの生産性を向上させるために、画面固有の GUI コントロールを提供します。詳しくは、173 ページの『第 7 章 画面イベントの処理』を参照してください。

- 一般的な要件は、複数のホスト画面からデータを収集して 1 つの出力ページに表示できること、複数のホスト画面にデータを供給する 1 つの入力ページを提供できることです。HATS では、このような操作を実行するための手法がいくつかあります。このような手法には、画面組み合わせ、画面カスタマイズ、変換、マクロ、グローバル変数、および統合オブジェクトなどの使用があります。使用する手法は、実行する操作とデータの場所によって異なります。最初に「画面組み合わせを作成」ウィザードを使用して、1 つのホスト画面に収まらないデータを組み合わせ、これを 1 つの出力ページに表示します。詳しくは、415 ページの『第 14 章 画面の結合』および 173 ページの『第 7 章 画面イベントの処理』を参照してください。

5. ホスト・アプリケーションでのユーザーのナビゲートを支援する単純なマクロを作成します。マクロを使用して、画面スキップ機能の実行、ユーザーに対して入力データを求めるプロンプトの表示、または 1 つ以上のホスト画面からのデータの抽出を実行できます。詳しくは、381 ページの『第 11 章 マクロとホスト端末』を参照してください。
6. Web アプリケーションの場合は統合オブジェクトを作成します。統合オブジェクトはホスト・アプリケーションとの対話をカプセル化する Java Bean であり、マクロから作成されます。次に、統合オブジェクトを使用して、ホスト・アプリケーションとの対話を実行するためのモデル 1、Struts、または JavaServer Faces (JSF) Web ページを作成します。各自の Java ビジネス・ロジックから統合オブジェクトを実行するか、または統合オブジェクトを拡張してホスト・アプリケーションへの Enterprise JavaBeans (EJB) または Web サービス・インターフェースを提供できます。詳しくは、403 ページの『第 13 章 統合オブジェクトの使用』を参照してください。

使用している HATS のバージョンを調べる方法

HATS Toolkit と各 HATS プロジェクトのバージョンを確認できます。

使用している HATS Toolkit のバージョンを判別するには、IBM Installation Manager のインストール済みパッケージを確認します。

1. 「スタート」メニューから「**すべてのプログラム**」>「**IBM Installation Manager**」>「**インストールされているパッケージの表示 (View Installed Packages)**」を選択します。
2. 「IBM Installation Manager - インストールされているオファリング (IBM Installation Manager - Installed Offerings)」ページに、インストールされているすべてのパッケージ (バージョンおよびビルド・レベルを含む) と機能がリストされます。

各 HATS プロジェクトのバージョン (保守レベルを含む) を確認できます。このバージョン情報は、プロジェクトごとに .serviceHistory.xml というファイルで保守されます。このファイルは、Web Content¥WEB-INF¥profiles フォルダー (Web プロジ

エクト)、profiles フォルダー (リッチ・クライアント・プロジェクト)、および ejbModule フォルダー (EJB プロジェクト) の HATS パースペクティブの「ナビゲーター」ビューで表示できます。

HATS .ear プロジェクトと HATS RCP ランタイム拡張プロジェクトのバージョン情報は、product.xml ファイルに維持されます。このファイルは、プロジェクトのルート・フォルダーの「ナビゲーター」ビューで表示できます。

問題判別情報の収集

HATS 関連の問題を判別するときにサポートが必要となる場合があります。 HATS サポート担当員用の問題判別情報を収集する方法については、「*HATS トラブルシューティング*」にある『問題判別情報の収集』を参照してください。

HATS に関する情報の検索場所

HATS をインストールする前に使用可能な HATS 情報を以下に示します。

- HATS インストール CD に HATS 資料が収録されています。インストール・ランチパッドから、以下の資料へのリンクを選択します。
  - *リリース情報*
  - *スタートアップ・ガイド (pdf)*
- HATS Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0) には、リリース情報、よくある質問 (FAQ)、API 参照情報、チュートリアル、WebFacing 情報があるほか、以下にリストされている資料の HTML 版と PDF 版も用意されています。翻訳済みの PDF 版も入手可能です。
  - *HATS スタートアップ・ガイド*
  - *HATS ユーザーと管理者のガイド*
  - *HATS Web アプリケーション・プログラマーズ・ガイド*
  - *HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*
  - *HATS 拡張マクロ・ガイド*
  - *HATS メッセージ*
  - *HATS トラブルシューティング*
  - *WebFacing アプリケーションの開発*
- HATS 製品の Web サイト (http://www.ibm.com/software/products/us/en/rhats) には、フィーチャーの概要、試用版のダウンロード、システム要件など、その他の製品情報があります。
- HATS HotSpot (https://www.ibm.com/developerworks/mydeveloperworks/groups/service/html/communityview?communityUuid=2ce1fd8d-d706-4afd-b9ef-9000ad21218d) には、リソース・ライブラリー、ディスカッション・フォーラム、ブログなどがあります。
- IBM Education Assistant Web サイト (http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/index.jsp) では、ナレーション入りのプレゼンテーション、デモンストレーション、チュートリアル、リソース・リンクを統合して、 IBM 製品を正常に使用できるように支援しています。目次の Host Access Transformation Services を検索してください。

HATS インストール後には、HATS インストール前に利用できる情報に加え、システムで以下の情報を利用できます。

- HATS の Web サイト、Knowledge Center、およびリリース情報へのリンクは Windows の「スタート」メニューにあります。
- Rational SDP メニュー・バーの「ヘルプ」メニューから、WebFacing を含む HATS 資料を利用できます。「**ヘルプ**」>「**ヘルプ目次**」を選択し、「目次」パネルで「**HATS アプリケーションの開発 (Developing HATS Applications)**」を選択します。
- HATS の開始ページのチュートリアルでは、HATS アプリケーションの開発過程について解説し、IBM Rational Developer for Power Systems Software™ 製品にインストールした場合の HATS Toolkit の IBM i フィーチャー用 IBM WebFacing Tool について説明しています。
- WebFacing Tool がインストールされている場合は、HATS の開始ページのチュートリアルと同様の情報が WebFacing の開始ページのチュートリアルで提供されます。
- コンテキストに依存したヘルプは、HATS ウィザードおよびエディター内のすべてのフィールドで使用できます。HATS Toolkit のヘルプを表示するには、F1 キーを押します。
- HATS プロジェクトの開発プロセスでは、キーポイントでヒントが表示されます。ヒントの表示と非表示を制御するには、HATS のプリファレンスを変更します。

第 2 章 HATS V9.5 へのマイグレーション

Host Publisher V4、HATS V5、V6、V7.0、V7.1、V7.5、V8.0、V8.5、V9.0 のいずれかのユーザーは、プロジェクトを HATS V9.5 にマイグレーションできます。

HATS V4 LE、HATS V5 LE、HATS V4 のプロジェクトを HATS V8.0、V8.5、V9.0、または V9.5 に直接マイグレーションすることはできません。それらのプロジェクトをマイグレーションするには、まず HATS の暫定リリース (V5、V6、V7、V7.1、V7.5 など) にマイグレーションし、その暫定リリースから HATS V8.0、V8.5、V9.0、または V9.5 にマイグレーションする必要があります。HATS プロジェクトのマイグレーションについて詳しくは、前のリリースの HATS の資料を参照してください。

V4.0 より前のバージョンの Host Publisher を使用している場合は、まず Host Publisher アプリケーションを Host Publisher V4.0 にマイグレーションする手順を実行する必要があります。http://www.ibm.com/support/entry/portal/overview/software/rational/rational_host_access_transformation_services にアクセスし、検索ストリングとして「How to migrate Host Publisher」を入力してください。詳しくは、 26 ページの『Host Publisher ユーザーのための HATS』を参照してください。

HATS のマイグレーション

前のリリースの HATS から HATS プロジェクトをマイグレーションするには、次の 2 つのステップのプロセスを実行します。

1. 以下のいずれかの方法で、プロジェクトをHATS V9.5 ワークスペースにインポートします。
   - 既存プロジェクトのルート・ディレクトリー、またはアーカイブ・ファイルを選択して、既存プロジェクトをインポートします。
   - コード・リポジトリー (CVS や IBM Rational ClearCase® など) からプロジェクトをチェックアウトします。
   - プロジェクトが含まれている Rational SDP V7.0 以降のワークスペースを Rational SDP V9.5 で開きます。
2. HATS マイグレーション・ウィザードを使用して、プロジェクトをマイグレーションします。

HATS プロジェクトのインポート

HATS V5 以降のリリースの HATS をインストールした場合は、プロジェクトを zip ファイルまたはプロジェクト交換ファイルにエクスポートし、続いて HATS V9.5 にインポートおよびマイグレーションすることができます。 HATS プロジェクトのエクスポートについて詳しくは、前のリリースの HATS の資料を参照してください。 HATS Web プロジェクトおよびリッチ・クライアント・プロジェクトの HATS V9.5 へのインポートについて詳しくは、 37 ページの『HATS Web プロジェクトのエクスポートとインポート』および 73 ページの『HATS リッチ・クライアント・プロジェクトのエクスポートとインポート』を参照してください。

© Copyright IBM Corp. 2003, 2015

ご使用の HATS プロジェクトを Rational ClearCase に保存した場合は、これらのプロジェクトを含むスナップショット・ビューを作成後、「**ファイル**」>「**インポート**」>「**一般**」>「**既存プロジェクトをワークスペースへ**」を使用してこれらのプロジェクトを HATS V9.5 ワークスペースにインポートする必要があります。このインポート中には、「**プロジェクトをワークスペースにコピーする (Copy projects into workspace)**」オプションを選択しないでください。ご使用の HATS V9.5 ワークスペースに各プロジェクトをインポート後、オプションで HATS V9.5 にそれらをマイグレーションすることができます。マイグレーション中に、更新の必要があるファイルは Rational ClearCase から自動的にチェックアウトされます。

Rational SDP ワークスペースのマイグレーション

Rational SDP ワークスペース・マイグレーション・ウィザードは、プロジェクトをワークスペースにインポートしたとき、いくつかのレベルの Rational SDP マイグレーションを実行する必要がある場合に表示されます。このウィザードによって、基本の Rational SDP プロジェクトのマイグレーションがいくつか実行され、プロジェクトのターゲットがサポートされないランタイムである場合は、サポートされるターゲット・ランタイムを選択するよう強制されます。HATS マイグレーション・ウィザードを実行する準備として、まず Rational SDP ワークスペース・マイグレーション・ウィザードを実行する必要があります。

注: Rational SDP V9.5 ワークスペースのマイグレーションでは、Rational SDP V7.0 以降で作成されたプロジェクトのみがサポートされます。 HATS V7.0 より前のプロジェクトをインポートすると、Rational SDP ワークステーションのマイグレーションは失敗し、エラー・メッセージが表示されます。ただし、HATS マイグレーションにより、プロジェクトを HATS V9.5 にマイグレーションすることは可能です。

HATS マイグレーション・ウィザードの使用

HATS パースペクティブに切り替える際に、前の HATS リリースから作成されたプロジェクトが HATS V9.5 ワークスペースにある場合は、マイグレーションする必要のある HATS プロジェクトがワークスペースに含まれていることを示すメッセージが表示されます。

「**問題**」ビューに表示されるエラーを調べる前に、HATS マイグレーション・ウィザードを使用してこのプロジェクトを HATS V9.5 にマイグレーションする必要があります。これは、マイグレーションを行うことにより、これらのエラーの一部またはすべてが除去される可能性があるためです。

HATS マイグレーション・ウィザードを使用するには、次の手順を実行します。

1. 「**HATS Projects**」 ビューで、プロジェクトを右クリックし、「**プロジェクトのマイグレーション**」を選択します。
2. マイグレーション対象として選択した HATS プロジェクトに関連付けられている前の HATS リリースのプロジェクトが他にもある場合、それらのプロジェクトも関連するすべての HATS .ear プロジェクトとともにマイグレーションされます。
3. 「**OK**」をクリックすると、マイグレーションが開始されます。
4. プロジェクトに関連する .ear プロジェクトがない場合は、マイグレーション後に .war プロジェクトを関連付ける必要があります。.war プロジェクトを .ear

プロジェクトと関連付ける方法は、 36 ページの『HATS Web プロジェクトを別の .ear ファイルに移動させる』を参照してください。

注:

1. マイグレーション後に、マイグレーション・レポートの詳細を確認してください。
2. マイグレーション中の HATS Web プロジェクトに、サポートされないターゲット・ランタイム (Rational SDP ワークスペース・マイグレーション・ウィザードを未実行の場合) またはサポートされない WebSphere ファセットが含まれている場合、HATS マイグレーション・ウィザードはターゲット・ランタイムおよびファセットを、Web プロジェクトに対してサポートされる最低の WebSphere Application Sever レベルに設定します。この変更を実行したことを示すメッセージが表示および記録されます。
3. マイグレーション後に、ターゲット・ランタイムが定義されていないことを示すエラーが表示される場合、プロジェクトのプロパティーを編集し、該当するターゲット・ランタイムを選択します。これを行うには、プロジェクトを右クリックし、「**プロパティー**」を選択します。「プロパティー」ウィンドウから、「**ターゲット・ランタイム**」を選択します。「ランタイム」リストから、該当するランタイムを選択します。
4. 前のリリースの HATS から HATS V9.5 にマイグレーションしたプロジェクトは、元に戻すことができません。

HATS 変換のマイグレーション

HATS プロジェクトを前のリリースの HATS から HATS V9.5 にマイグレーションする際には、プロジェクト内の変換もマイグレーションされます。変換に対して行われる変更は、マイグレーション元の HAT のリリースによって異なります。すべての変換は、マイグレーションする前に MigrationBackup フォルダーに保存されます。

特別なマイグレーションの考慮事項

自動拡張

前のバージョンの HATS で lxgwfunctions.js ファイルに行 `turnAutoTabOn();` を追加することによって自動拡張機能を使用可能にした場合は、プロジェクト設定エディターの「**その他**」タブの「クライアント設定」セクションで「**自動フィールド進行を使用可能にする**」設定を使用する必要があります。詳しくは、 142 ページの『クライアント設定』を参照してください。

バックアップ・ファイル

HATS マイグレーションでは、プロジェクト・ディレクトリー内に **MigrationBackup** というバックアップ・フォルダーが作成されます。このフォルダーには、マイグレーション・プロセスによって上書きされる前の、古いプロジェクトのファイルのコピーが格納されています。これらのファイルは保存されるため、新規 HATS V9.5 ファイルと比較およびマージすることができます。問題が発生してもあまり支障はありません。これらのファイルはアプリケーションではもう使用されません。保存されたすべてのファイルの比較とマージが済んだら、**MigrationBackup** フォルダーを削除して構いません。

注: MigrationBackup フォルダーを削除する前に、マイグレーション済みのプロジェクトを WebSphere Application Server に公開して、動作をテストできます。ただし、タスク・リストに示されている MigrationBackup フォルダーが原因のエラーによって、 WebSphere Application Server への公開の際に問題が発生することがあります。この問題は、プロジェクトから MigrationBackup フォルダーを除去するか、エラーを含んだアプリケーションをサーバーに公開することを許可することによって回避できます。エラーを含んだアプリケーションのサーバーへの公開を許可するには、 Rational SDP メニュー・バーから「**ウィンドウ**」>「**設定**」>「**サーバー**」>「**WebSphere Application Server**」を選択し、「**エラーを含むアプリケーションのサーバーへの公開を許可する (Allow applications containing errors to be published on a server)**」を選択します。

ボタン・ウィジェット

HATS V7.0.0.2 より前のバージョンで作成されたプロジェクトをマイグレーションする場合に、「**前景色を使用可能にする**」オプションをインプリメントすることを選択し、blacktheme.css スタイル・シートを使用するテンプレートを使用する場合、以下のいずれかの方法で blacktheme.css スタイル・シートを手動で更新する必要があります。

1. 次の行を input.HATSBUTTON 宣言から除去します。

   ```
   color: lime;
   ```

   これにより、前景色をファンクション・キーでレンダリングできるようになります。ただし、HATS ウィジェットで生成された他のボタンはデフォルトの色でレンダリングされるようになります。

2. 各色ごとに色に関連する CSS 宣言を結合します。例えば、以下のように変更します。

   ```
   .HBLUE {
          color: #3c9dff;
   }
   input.HBLUE {
          white-space: normal;
          letter-spacing: normal;
   }
   ```

   to

   ```
   .HBLUE, input.HBLUE {
           color: #3c9dff;
           white-space: normal;
           letter-spacing: normal;
   }
   ```

   各色ごとにこの変更を繰り返します。

blacktheme.css スタイル・シートを手動で編集する以外に、新規のダミー・プロジェクトを作成し、そのプロジェクトからご使用のプロジェクトにテーマ CSS ファイルをコピーする方法もあります。この場合、CSS ファイルに加えられていた変更はすべて上書きされることに注意してください。

EJB Access Bean

EJB Access Bean を持つ既存の HATS V5 プロジェクトがある場合、「HPubReqCompleteEvent を解決できません。」で始まるコンパイルまたはランタイム・エラーが発生する可能性があります。このエラーを訂正するには、EJB Access Bean を再生成します。

注: HATS EJB アプリケーション・サポートは、HATS V9.5 では推奨されません。HATS EJB アプリケーションのサポートは現在は継続されていますが、IBM は製品の後続リリースでこの機能を除去する権利を持っています。以下の代替方法があります。

- Web サービスを使用して統合オブジェクトにアクセスする。詳しくは、「*HATS Web アプリケーション・プログラマーズ・ガイド*」の『Web サービスの開発』を参照してください。
- カスタム EJB Bean を作成して統合オブジェクトにアクセスする。詳しくは、「*HATS Web アプリケーション・プログラマーズ・ガイド*」の『EJB コンテナーでの統合オブジェクトの使用 (独自の EJB から)』を参照してください。

フィールド・ウィジェット

フィールド・ウィジェットに「**モノスペース・フォントを使用してレンダリング**」設定を使用し、かつプロジェクトが HATS V7.0 以前のリリースで作成されている場合、CSS ファイルを以下のように更新する必要があります。

1. 次のクラスを whitetheme.css、graytheme.css、monochrometheme.css、tantheme.css、および blacktheme.css ファイルに追加します。

   ```
   .HF {
   font-family: courier new, monospace;
   }
   ```

2. すべての CSS ファイルのすべての H-color (HATS カラー) クラスから、`font-family: monospace` を除去します。

グローバル規則

新しい設定 **enforceImmediacy** は、HATS V5 から HATS V9.5 にインポートされたすべてのグローバル規則に追加されます。この設定が true の場合、「**周囲のテキストによって入力フィールドを検出**」パターンの「**変換**」オプションで「**最も近い入力フィールドのみ**」が選択されていると、グローバル規則は HATS V5 の場合と同様の振る舞いをします。この設定が false (デフォルト) の場合、グローバル規則は HATS V9.5 の場合と同様の振る舞いをします。

新しく定義したグローバル規則の振る舞いを HATS V5 のグローバル規則と同様にするには、**enforceImmediacy** 設定を追加して、これを true に設定する必要があります。これを行うには、「プロジェクト設定」エディターの「**ソース**」 タブで、以下の設定を globalRules componentSettings タグに追加します。

```
<setting name="enforceImmediacy" value="true"/>
```

グローバル変数

HATS V7 よりも前は、「**グローバル変数を入力ボックスで要求する**」オプションを使用して変換へ挿入されたプロンプトに入力されたグローバル変数の値は、一部の文字が適切にエスケープされなかったために切り捨てられる場合がありました。

この HATS バージョンでは、変換によってプロンプトが出されるグローバル変数の値は、切り捨てを防ぐために適切にエスケープされます。「**グローバル変数値を静的なテキストとして表示**」オプションを使用して変換に挿入されるグローバル変数は、この変更による影響を受けません。HATS V7 より前の HATS のバージョンで変換に挿入されたグローバル変数のプロンプトは、マイグレーション中に自動的には更新されないため、この変更による影響を受けません。

HTTP 圧縮

HATS V5.0.x、V6.0、および V6.0.1 からマイグレーションしたプロジェクトで HTTP 圧縮に HATS サポートを使用する場合は、圧縮フィルターを Web デプロイメント記述子ファイル (web.xml) に手動で追加する必要があります。圧縮フィルターを web.xml ファイルに追加するには、以下の手順を実行します。

1. HATS Toolkit から、HATS パースペクティブの「**ナビゲーター**」ビューに切り替えます。
2. プロジェクトの Web Content¥WEB-INF フォルダーにある web.xml ファイルを開きます。
3. 「**ソース**」タブをクリックして、このファイルのソースを表示します。
4. 最後に定義されたサーブレット・マッピング (最後の `</servlet-mapping>` を検索します) の後ろに、以下のステートメントをコピーします。

```
<filter>
   <description>Provides compression for output from HATS entry
                servlet</description>
   <display-name>CompressionFilter</display-name>
   <filter-name>CompressionFilter</filter-name>
   <filter-class>com.ibm.hats.runtime.filters.CompressionFilter</filter-class>
</filter>
```

5. 最後に定義されたフィルター・マッピング (最後の `</filter-mapping>` を検索します) の後ろに、以下のステートメントをコピーします。

```
<filter-mapping>
   <filter-name>CompressionFilter</filter-name>
   <servlet-name>EntryServlet</servlet-name>
   <dispatcher>ERROR</dispatcher>
   <dispatcher>FORWARD</dispatcher>
   <dispatcher>INCLUDE</dispatcher>
   <dispatcher>REQUEST</dispatcher>
</filter-mapping>
<filter-mapping>
   <filter-name>CompressionFilter</filter-name>
   <url-pattern>/</url-pattern>
</filter-mapping>
<filter-mapping>
   <filter-name>CompressionFilter</filter-name>
   <url-pattern>/index.jsp</url-pattern>
</filter-mapping>
```

6. ファイルを保存します。

注: サーバー上でこのプロジェクトを既に実行している場合は、WebSphere Application Server が web.xml ファイルへの変更を取得できるようにアプリケーションを再公開する必要があります。

IBM ライセンス管理ツール

HATS V9.5 にマイグレーションされた HATS アプリケーション、WebFacing アプリケーション、および HATS/WebFacing 接続アプリケーションを実稼働環境に再デ

プロイして、IBM ライセンス管理ツール (IBM License Metric Tool や IBM Tivoli® Asset Discovery for Distributed など) による新規 HATS V9.5 シグニチャー・ファイルに対するサポートを取得する必要があります。

WebFacing Web アプリケーション: IBM ライセンス管理ツールは、サポート対象の WebSphere Application Server に Enterprise Archive (EAR) としてデプロイされている Web アプリケーションを検出します。 WebFacing Web プロジェクトは、関連する EAR から独立して作成されるため、HATS V9.5 シグニチャー・ファイルを手動で組み込む必要があります。以下の手順を実行して、WebFacing Web アプリケーションが適切に検出されるようにします。

1. WebFacing プロジェクトの関連 EAR の直下に `itlm` という名前のフォルダーを作成します。
2. HATS プラグイン・ディレクトリーで `Host_Access_Transformation_Services-9.0.0.swidtag` という名前のシグニチャー・ファイルを見つけます。

   `<`*`shared_install_directory`*`>¥plugins¥com.ibm.hats_`*`nnn`*`¥`

   ここで、*shared_install_directory* は IBM Installation Manager を使用して HATS オファリングをインストールした共用リソース・ディレクトリー、*nnn* は HATS のバージョンおよびビルド・レベルです。
3. プラグインのシグニチャー・ファイルを上記で作成した `itlm` フォルダーにコピーします。
4. プロジェクトを EAR としてエクスポートし、再デプロイします。
5. これで、IBM ライセンス管理ツールは、ご使用のアプリケーションを検出できるようになります。

旧 WebFacing プロジェクトで検出を有効にし、そのプロジェクトを V9.5 にマイグレーションする場合は、再デプロイを行う前に、既存のシグニチャー・ファイル (例えば、前の WebFacing プロジェクトが V7.0 であれば `WDHT0700.sys2`) をプロジェクトの関連 EAR から除去し、V9.5 のシグニチャー・ファイル (`Host_Access_Transformation_Services-9.0.0.swidtag`) を追加する必要があります。

Java 2 セキュリティー

HATS V9.5 へのマイグレーション中に、WebSphere Application Server Java 2 セキュリティー was.policy ファイルが上書きされます。 HATS V9.5 より前のプロジェクトで was.policy ファイルをカスタマイズした場合は、マイグレーション後にそのファイルを再度カスタマイズする必要があります。

キーボード・マッピング

以前のバージョンの HATS ホスト端末では、Pause キーはホストの [clear] アクションにマップされていました。 HATS V7 以降のホスト端末では、Esc キーがホストの [clear] アクションにマップされます。

HATS V6 より前のバージョンの HATS では、双方向セッションで画面を反転させるために Ctrl+R が使用されていました。HATS V6 以降では、Ctrl+R は双方向セッションと非双方向セッションの両方のホストの RESET にマップされ、Alt+Enter は双方向セッションの反転にマップされています。

リッチ・クライアント・アプリケーション

Lotus Notes キーボード・バインディング:

Lotus Expeditor をターゲットとする HATS V7.5.1 より前のリッチ・クライアント・プロジェクトを HATS V9.5 にマイグレーションする場合は、Lotus Notes 用のキーボード・マッピングがプロジェクトに追加されます。

起動構成:

リッチ・クライアント・プロジェクトのマイグレーション後、それをテスト環境で起動できない場合は、テスト環境で使用する新規起動構成を作成します。これを行うには、「HATS プロジェクト」ビューで、プロジェクトを右クリックし、「**実行**」または「**デバッグ**」のいずれかを選択します。「実行」(または「デバッグ」) 構成ウィンドウが表示されます。プロジェクトが Eclipse RCP をターゲットとする場合は、「**Eclipse アプリケーション**」を右クリックし、「**新規**」を選択します。プロジェクトが Lotus Notes または Lotus Expeditor をターゲットとする場合は、「**クライアント・サービス**」を右クリックし、「**新規**」を選択します。必要な場合は、起動構成の名前と場所を変更します。このウィンドウから古い起動構成を削除することもできます。「**実行**」 (または 「**デバッグ**」) をクリックして、テスト環境のプロジェクトを起動します。

テーマ:

「**従来型端末エミュレーター**」テーマを使用する HATS V7.0 リッチ・クライアント・アプリケーションでは、マイグレーション後に「**矢印キー・ナビゲーション**」設定が使用可能になります。

テンプレートおよび変換での米国英語以外のストリング:

HATS V7.0 または V7.0.0.1 を使用して作成された HATS リッチ・クライアント・プロジェクトのテンプレートおよび変換に含まれている米国英語以外のストリングは、プロジェクトがフィーチャーとしてエクスポートされた場合に、正しくコンパイルされません。この問題に対処するには、プロジェクトを HATS V9.5 にマイグレーションした後で、プロジェクトのルート・フォルダーにある build.properties ファイルを編集します。build.properties ファイルの一番下に、次の行を追加します。

```
javacCustomEncodings.library.jar = src/rcpproject/templates/[UTF-8],
 src/rcpproject/transformations/[UTF-8]
```

ここで、*rcpproject* は リッチ・クライアント・プロジェクトの名前です。

ランタイムの使用可能化

HATS V9.5 ランタイムを実稼働環境用に完全に使用可能にするには、「ライセンス設定」ウィザードを使用してライセンス設定を指定してください。この作業は、ランタイムが前のバージョンの HATS で完全に有効であったプロジェクトでも、行う必要があります。ライセンス設定の指定については、「*HATS スタートアップ・ガイド*」の『HATS ランタイムおよびライセンス設定を有効にする』を参照してください。

注: ランタイムが完全に使用可能ではない HATS Web アプリケーションは、2 つのランタイム・ホスト接続のみに制限されます。ランタイムが完全に使用可能

ではない HATS リッチ・クライアント・アプリケーションは、ローカル・テスト環境ではホスト接続が無制限に許可されますが、デプロイされた実稼働環境ではホスト接続は許可されません。

Secure Sockets Layer (SSL)

マイグレーション中に、HATS V9.5 は SSL 証明書クラス・ファイル `CustomizedCAs.class` を、名前が *appname*`-CustomizedCAs.p12` でパスワードが `hats` の PKCS12 鍵ストア・ファイルに変換します。 *appname* はプロジェクト名です。マイグレーション後、証明書管理ツール (IBM 鍵管理、または iKeyMan とも呼ばれる) を使用して新規鍵ストア・ファイルのパスワードを変更し、接続エディターを使用してそれらのファイルを検証する必要があります。詳しくは、 166 ページの『セキュリティー』を参照してください。

テスト・モード

HATS V7 からは、「サーバーで実行」(Web プロジェクト用) および「実行」(リッチ・クライアント・プロジェクト用) のテスト・モードを使用して、実稼働環境に展開するランタイム設定を変更およびテストすることができます。このモードでのテスト中にランタイム設定に変更が加えられた場合、その変更は保持され、HATS アプリケーションを実稼働環境に展開するときに有効になります。

「サーバーでデバッグ」(Web プロジェクト用) および「デバッグ」(リッチ・クライアント・プロジェクト用) のテスト・モードを使用して、実稼働環境に展開する設定を変更せずにランタイム設定を変更およびテストすることができます。

Web プロジェクトのランタイム設定の変更の詳細については、 447 ページの『HATS Toolkit での管理コンソールの開始』を参照してください。リッチ・クライアント・プロジェクトのランタイム設定の変更の詳細については、 93 ページの『HATS リッチ・クライアント・アプリケーションの管理』を参照してください。

URL ホスト・コンポーネント

HATS V7 より前のバージョンでは、URL ホスト・コンポーネントは入力フィールドと保護フィールドの両方の中を検索していました。HATS V7 からは、URL ホスト・コンポーネントは保護された非隠しフィールド内のみを検索します。

Web 高速ログオン (WEL)

HATS Web プロジェクトが関連 EAR を伴わずに HATS V9.5 にマイグレーションされると、Web 高速ログオン (WEL) は正しく機能しません。 Web プロジェクトの WEL 構成情報は、その関連 EAR プロジェクト内に保持されています。関連 EAR プロジェクトがインポートおよびマイグレーションされないと、この情報は失われます。

Web プロジェクトと EAR プロジェクトをアーカイブ・ファイルとしてワークスペースにインポートし、Web プロジェクトをマイグレーションすると、関連 EAR プロジェクトもマイグレーションされ、WEL 構成情報は失われません。

HATS V7 からは、HATS DCAS/RACF/JDBC Credential Mapper プラグインと証明書ベースの DCAS/RACF Credential Mapper プラグインの両方で、DCAS サーバー

へのセキュア接続に Java Secure Socket Extension (JSSE) が使用されます。このため、プラグインの初期化パラメーターに以下の変更を行うことを検討してください。

CMPI_DCAS_KEYRING_FILE

このパラメーターは推奨されないため、使用しないでください。ただし、使用した場合、このパラメーターは HATS V9.5 で非推奨パラメーター CMPI_DCAS_KEYRING_PASSWORD とともにサポートされ、鍵リング・タイプは pkcs12 であると想定されます。代わりに CMPI_DCAS_TRUSTSTORE を使用してください。詳しくは、487 ページの『初期化パラメーター』を参照してください。

このパラメーターは鍵リング・データベースを指定します。DCAS クライアント証明書のほか、DCAS サーバーの証明書にもアクセスできるようにするには、鍵リングを指定する必要があります。この証明書は、DCAS サーバーとのクライアント認証されたセキュア接続を確立します。このパラメーターは、使用する鍵リングに対するファイル参照です。DCAS プラグインは、DCAS クライアントです。鍵リング・ファイルは .ear ファイルに格納する必要があります。

CMPI_DCAS_KEYRING_PASSWORD

このパラメーターは推奨されないため、使用しないでください。ただし、使用した場合、このパラメーターは HATS V9.5 で非推奨パラメーター CMPI_DCAS_KEYRING_FILE とともにサポートされ、鍵リング・タイプは pkcs12 であると想定されます。代わりに CMPI_DCAS_TRUSTSTORE_PASSWORD を使用してください。詳しくは、487 ページの『初期化パラメーター』を参照してください。

このパラメーターは、鍵リング・データベースのパスワードを指定します。

Host Publisher ユーザーのための HATS

IBM WebSphere Host Publisher の使用に慣れている場合は、HATS でプロジェクトを開発するためにアプローチを調整する必要があります。ここでは、プロジェクトを開発する際の HATS と Host Publisher とのアプローチの主な違いについて説明します。

- HATS サーバーのインストールはありません。HATS Toolkit でエクスポートする際に、*HATS ランタイム* と呼ばれる HATS 実行可能コードが各 HATS .ear に組み込まれます。HATS プロジェクトでは、単一の .ear ファイルに複数の .war ファイルを組み込むこともできます。
- Host Publisher は、ホスト画面の変換やユーザーへの表示を行いません。ホスト・アプリケーションから特定の情報を検索してそれを表示します。この機能を実行する HATS プロジェクトを作成できますが、HATS の基本機能は、ホスト画面を変換してそれをユーザーに表示することです。また、HATS プロジェクトを拡張して画面を結合またはスキップしたり、複数のホスト・アプリケーションのデータを結合したりすることもできます。HATS では、新規のプロジェクトを徐々に作成していくことができ、プロジェクトへの後からの追加も容易です。
- Host Publisher プロジェクトの機能は明確に構成されていました。ユーザーが実行できるホストとの対話を指定することによって、ユーザーの動作が制御されていました。HATS で行われる規則に基づいたアプローチでは、HATS プロジェクト

が、ユーザーがアクセスするホスト画面を変換し、ユーザーとそれらの画面との対話を処理するための規則を設定します。Host Publisher プロジェクトでユーザーがホストとの限定された対話を実行できるように設定されていた部分に関して、HATS プロジェクトは規則を適用し、ユーザーが実行する一連の対話を処理します。HATS では、ユーザーとホストとの対話の自由度を定義できます。対話の範囲は、フリー・フォームのホスト・エミュレーターから、完全に指示された画面ナビゲーションにまで渡ります。

- Host Publisher では、ホストとの対話がすべて、統合オブジェクトにカプセル化されたマクロ内で実行されます。HATS では、数多くの目的にマクロを使用できますが、ホスト画面の基本変換には必要ありません。Host Publisher では、統合オブジェクトに、ホスト・アプリケーションからデータを抽出するためのデータ・マクロだけでなく接続マクロと切断マクロも必要です。HATS では、HATS アプリケーションが既に接続されているホスト上でマクロを実行できます。このように、HATS では、統合オブジェクトに常に接続マクロと切断マクロが必要であるとは限りません。

注: 変更済みのテンプレートを使用して Host Publisher の統合オブジェクトを作成した場合は、HATS 統合オブジェクト・テンプレートに同じ変更を加え、マクロで統合オブジェクトを再生成することによって、統合オブジェクトを再作成する必要があります。統合オブジェクトをカスタマイズするテンプレートの使用方法の詳細については、「*HATS Web アプリケーション・プログラマーズ・ガイド*」を参照してください。

Host Publisher バージョン 4.0 アプリケーションをインポートする場合 (Host Publisher バージョン 4.0.1 アプリケーションではなく) は、次の手順を実行します。

- リモート統合オブジェクト (RIO) はサポートされなくなりました。すべての Host Publisher バージョン 4 アプリケーションに RIO サーブレットが含まれています。RIO サポートを含む Host Publisher V4 アプリケーションをインポートすると、RIO サーブレットは削除され、それについてのマイグレーション・メッセージが表示されます。RIO を使用してリモートで統合オブジェクトにアクセスしていた場合は、Web サービスを使用するようにアプリケーションを変更する必要があります。詳細については、「*HATS Web アプリケーション・プログラマーズ・ガイド*」を参照してください。
- EJB 1.0 サポートを含む Host Publisher アプリケーションをインポートしようとすると、アプリケーションに EJB 1.0 サポートが含まれているというエラー・メッセージが表示されます。アプリケーションをインポートするには、Host Publisher Studio に戻り、EJB 1.1 仕様レベルを使用して EJB Access Bean を再生成し、Host Publisher Application Integrator を使用してアプリケーションを再度パッケージする必要があります。

Host Publisher バージョン 4 からのマイグレーション

現在 Host Publisher V4 を使用している場合は、プロジェクト全体をマイグレーションすることも、既存のプロジェクトの統合オブジェクトだけをマイグレーションすることもできます。Host Publisher の .ear ファイルをインポートした場合、統合オブジェクトは .jar ファイル内にパッケージされたままになります。

統合オブジェクトが Web モジュール内に含まれていた場合、.jar ファイルは WEB-INF/LIB ディレクトリー内にあります。統合オブジェクトが EJB モジュール内に含まれている場合、.jar ファイルは imported_classes¥IntegrationObject ディレクトリー内にあります。EJB Access Beans も、Web モジュールの WEB-INF/LIB ディレクトリーの .jar ファイル内にパッケージされたままになります。

.ear をインポートし、統合オブジェクトを作成したマクロを変更すると、インポートされた元の統合オブジェクト .jar と Java ソースが組み込まれます。HATS 用の統合オブジェクトを再生成したら、インポート済み .jar ファイルは削除して構いません。

Host Publisher V4 からマイグレーションする際の考慮事項を以下に示します。

- Host Publisher EJB アプリケーションをインポートする場合、アプリケーションを実行しようとすると、エラーが発生します。com/ibm/HostPublisher/IntegrationObject/HPubReqCompleteEvent に関する java.lang.NoClassDefFoundError を受け取ります。このエラーを訂正するには、EJB Access Bean を再生成して、Web アプリケーションの WEB-INF¥LIB ディレクトリーにある EJB Access Bean .jar ファイルを削除してください。さらに、名前の不一致に関するコンパイル・エラーを修正するには、JavaServer Pages または Java コードで新規に生成される名前を参照するように EJB Access Bean 名を更新する必要があります。
- Web サービス・サポートを使用する統合オブジェクトまたは EJB Access Beans がある場合は、Web サービスを再生成する必要があります。Web サービスの再生成の詳細については、「*HATS Web アプリケーション・プログラマーズ・ガイド*」を参照してください。
- Database Access 統合オブジェクトは、HATS V6 以降では推奨されません。Database Access 統合オブジェクトを含む Host Publisher アプリケーションをインポートすると、その統合オブジェクトは .jar ファイルに保管されます。ただし、HATS V6 以降で新規 Database Access 統合オブジェクトを作成することはできません。Rational SDP のリレーショナル・データベース・ツールを使用して、使用するリレーショナル・データベースにアクセスすることができます。
- 定義済みの XML ゲートウェイ接続はマイグレーションされません。デフォルトの変換を使用して HATS アプリケーションを定義して、XML ゲートウェイ接続と同じ機能を実行することができます。
- XML Legacy Gateway SDK は、HATS では使用できません。
- Host Publisher 高速ログオンは、HATS ではサポートされていません。高速ログオンを使用するように設定された統合オブジェクトは、HATS Web 高速ログオン (WEL) を使用するように設定する必要があります。高速ログオン機能 (ELF) のサポートが組み込まれた統合オブジェクトをインポートすると、統合オブジェクトはエラーなしでマイグレーションされますが、ELF がサポートされないこと、および同等の機能を実行するには WEL を使用する必要があることを示す警告が発行されます。WEL アクションを使用するように接続マクロを再記録し、統合オブジェクト接続を新規 WEL マクロと関連付けて、WEL 構成を行う必要があります。詳細については、481 ページの『Web 高速ログオン (WEL) の使用』を参照してください。これを実行しないと、統合オブジェクトは意図したとおりに実行されません。
- 暗号化されたユーザー・リストが HATS でサポートされるようになりました。ただし、ユーザーが選択した暗号鍵を使用してのユーザー・リストの暗号化は

HATS ではサポートされていません。ユーザーが選択した暗号鍵で暗号化された Host Publisher ユーザー・リストをインポートするには、ユーザー・リストの暗号化に使用された鍵を提供する必要があります。ユーザー・リストはインポート時に暗号化解除されます。インポートした後に、HATS で使用するためにユーザー・リストを暗号化することができます。暗号化するには、接続エディターを開いて、「**ユーザー・リスト**」タブをクリックし、「**ユーザー・リスト・プロパティーの暗号化**」を選択します。「ユーザー・リスト」タブでは、「ユーザー ID」、「説明」、および「パスワード」フィールドのみ編集することができます。Host Publisher ユーザー・リストに追加したユーザー定義プロパティーを編集する必要がある場合、まず「ユーザー・リスト」タブで「**ユーザー・リスト・プロパティーの暗号化**」の選択をクリアしてから、「**ソース**」タブをクリックして、プロパティー値を編集する必要があります。スキーマ・エレメントは編集しないでください。ユーザー・リストを再度暗号化する場合は、「**ユーザー・リスト**」タブをクリックし、「**ユーザー・リスト・プロパティーの暗号化**」を選択して、ファイルを保存します。ユーザー・リスト・データの暗号化を選択したかどうかには関係なく、機密データの保護には、ファイル・システム・セキュリティーと物理的セキュリティーの手段を使用する必要があります。

- 変更されたテンプレートを使用して作成された統合オブジェクトがある場合、これらの統合オブジェクトを含む Host Publisher アプリケーションをマイグレーションするには、以下の手順を実行します。
  1. Host Publisher EAR をインポートします。変更されたテンプレートを含む統合オブジェクトをインポートしているという警告が表示されます。
  2. インポートした統合オブジェクトのいずれかに、テーブルとして抽出されたデータが含まれている場合は、それらの個々の統合オブジェクトをインポートします。
  3. HATS に付属している統合オブジェクト・テンプレートを変更し、Host Publisher で定義したのと同じカスタマイズを行います。詳細については、「*HATS Web アプリケーション・プログラマーズ・ガイド*」を参照してください。

「Host Publisher EAR のインポート」ウィザード

Host Publisher バージョン 4 プロジェクトは、新規 HATS プロジェクトとして HATS にインポートすることもできますし、既存の HATS プロジェクトの一部としてインポートすることもできます。

1. 「**ファイル**」>「**インポート**」>「**HATS**」>「**Host Publisher V4 EAR**」をクリックして、「Host Publisher EAR ファイルのインポート」ウィザードを開きます。
2. 「**EAR ファイル名**」フィールドに、インポートする EAR ファイルを指定します。
3. 新規 HATS プロジェクトと既存の HATS プロジェクトのどちらに対して EAR ファイルを使用するかを決めます。
4. 「**オプション**」セクションでは、チェック・ボックスで「**警告なしでリソースを上書き**」を設定できます。
5. Host Publisher で暗号化を使用している場合は、「**ユーザー・リスト暗号鍵**」テキスト・ボックスに暗号鍵を入力できます。
6. 「**完了 (Finish)**」をクリックする。

Host Publisher 統合オブジェクトのインポート

個々の Host Publisher 統合オブジェクトを既存の HATS プロジェクトにインポートすることもできます。

1. 「**ファイル**」>「**インポート**」>「**HATS**」>「**Host Publisher V4 統合オブジェクト**」をクリックして、「Host Publisher 統合オブジェクトのインポート」ウィザードを開きます。
2. 「**Host Publisher 統合オブジェクト・ディレクトリー**」フィールドに、インポートする統合オブジェクトの場所を指定します。
3. 「**インポートする統合オブジェクトを選択してください**」 フィールドで、インポートする統合オブジェクトを選択します。
4. 「**HATS プロジェクトを選択してください**」 ドロップダウン・ボックスで、HATS プロジェクトを選択します。
5. 「**警告なしでリソースを上書き**」または「**インポートした統合オブジェクトを再生成します**」 チェック・ボックスをオンにすることができます。

   注: チェーニング統合オブジェクトがある場合は、必ず「**インポートした統合オブジェクトを再生成します**」 オプションをオンにしてください。これによって、チェーニングされた情報が確実に保持されます。チェーニングされた統合オブジェクトをインポートした後、状態情報は Java ソースに置かれ、「チェーニング統合オブジェクトの作成」ウィザードの使用時には情報が事前入力されます。チェーニングされた統合オブジェクトの詳細については、405 ページの『統合オブジェクトのチェーニング』を参照してください。
6. 「**完了 (Finish)**」をクリックする。

注: 統合オブジェクトをインポートすると、統合オブジェクトに関連する接続設定が HATS プロジェクトに追加されます。

J2EE マイグレーション

HATS プロジェクトの J2EE サポートのレベルをマイグレーションするには、以下の手順を実行してください。

1. Rational SDP で、「ナビゲーター」ビューに切り替えます。
2. HATS プロジェクトを含む .ear プロジェクトを右クリックして、「**Java EE**」>「**仕様アップグレード・ウィザード (Specifications Upgrade Wizard)**」を選択します。
3. 「ようこそ」ページのパネルで、「**次へ**」をクリックします。
4. 「エンタープライズ・アプリケーション」パネルで、適切な J2EE バージョンを選択して、「**すべてのモジュール・プロジェクトをマイグレーションする**」ボックスをクリアします。

   注: このボックスをクリアすると、マイグレーションするプロジェクトに個別にチェック・マークを付けることができます。.ear プロジェクトに HATS EJB プロジェクトが含まれている場合、.ear プロジェクトを正常にマイグレーションするためには、その HATS EJB プロジェクトを選択しないでください。

5. 「EJB モジュール」パネル (表示された場合) では、HATS EJB プロジェクトを選択しないでください。
6. 「Web プロジェクト」パネルで、すべての HATS Web プロジェクトを選択します。
7. 「**完了**」をクリックします。

第 3 章 HATS Web アプリケーションの開発および展開

HATS アプリケーションのタイプには、Web、ポートレット、EJB、リッチ・クライアント、および管理コンソールがあります。この章では、HATS Web アプリケーションについて説明します。他のタイプのアプリケーションについては、以下の章を参照してください。

- EJB、「*HATS Web アプリケーション・プログラマーズ・ガイド*」 の章『HATS EJB アプリケーションの作成と使用』
- リッチ・クライアント, 65 ページの『第 4 章 HATS リッチ・クライアント・アプリケーションの開発とデプロイ』
- 管理コンソール, 443 ページの『第 18 章 HATS 管理コンソールの使用』

また、HATS Web アプリケーションを WebFacing アプリケーションと関連付け、両方を一緒にパッケージ化して、単一エンタープライズ・アプリケーションで相互運用することができます。詳しくは、 473 ページの『第 20 章 WebFacing と HATS』を参照してください。

HATS アプリケーションは、HATS Toolkit を使用して HATS プロジェクトから作成されます。新規の HATS プロジェクトを作成すると、HATS アプリケーション・ファイルを編成するために利用できる一組のフォルダーが作成されます。以下にデフォルト・プロジェクトの例を示します。最上位のフォルダーの名前は、作成するプロジェクトに付けた名前と同じです。このフォルダーの中には、HATS プロジェクトで定義したオブジェクトを含む他の上位フォルダーが入っています。一部のフォルダーは、特定のオブジェクトが作成されるまで表示されません。

© Copyright IBM Corp. 2003, 2015

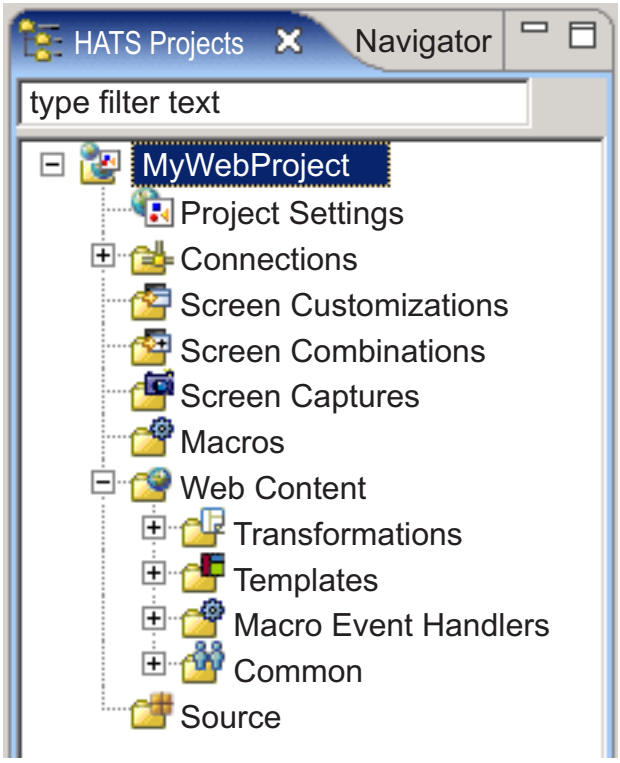


図 2. 「HATS Web プロジェクト」ビュー

HATS プロジェクトのセットアップ方法に応じて、これらのフォルダーの一部またはすべてが「**HATS プロジェクト**」ビューに表示されます。「**HATS プロジェクト**」ビューに表示するフォルダーや、ファイル拡張子の非表示を指定することもできます。詳しくは、149 ページの『HATS プリファレンスの使用』を参照してください。

注: HATS プロジェクトのタイプによって、表示されるフォルダーは異なります。例えば、HATS EJB プロジェクトのディレクトリー・ツリーには、「**画面カスタマイズ**」、「**画面キャプチャー**」、および「**Web コンテンツ**」フォルダーはありません。

プロジェクトのファイルの編成をわかりやすくするために、これらの上位サブフォルダーの内部にもフォルダーを作成することができます。例えば、プロジェクト用の画面キャプチャーを作成するときに、「**画面キャプチャー**」フォルダーの下に、取り込んだ画面を編成してグループ化するためのフォルダーを作成することができます。フォルダーを作成するには、ツリー内の上位フォルダーの 1 つを右マウス・ボタンでクリックし、「**新規 HATS**」>「**フォルダー**」を選択します。ファイルを別のフォルダーに移すには、該当ファイルを右マウス・ボタンでクリックして「**移動**」を選択するか、ドラッグ・アンド・ドロップ方式を使用します。

注: サブフォルダーには制限事項があります。変換ファイルおよびテンプレート・ファイルは、同じレベルのサブフォルダーに格納する必要があります。あるレベルのサブフォルダー (¥transformations¥Callup¥ など) にある変換を使用するには、この変換と組になるテンプレートが同じレベルのサブフォルダー (¥templates¥Callup¥ など) になければなりません。

HATS パースペクティブの「**ナビゲーター**」**ビュー**に進むと、HATS プロジェクトをチーム環境で共有できます。プロジェクトを右クリックして、「**チーム**」>「**プロジェクトの共用**」を選択します。リストからリポジトリー・タイプを選択し、「**次へ**」をクリックします。Rational SDP は、いくつかのリポジトリーをサポートしています。詳しくは、Rational SDP の資料を参照して「リポジトリー」を検索してください。

注: HATS プロジェクトでバージョン管理システムを使用する場合は、`resourceUpdate.sts` ファイルを無視するようにバージョン管理システムを設定します。このファイルは、ツールキット内でプロジェクトをテストするときに自動的に生成されます。このファイルはバージョン管理では不要であり、バージョン管理下で HATS プロジェクトを設置する前に安全に無視したり削除したりできます。

このファイルをバージョン管理から除外するには、「**ウィンドウ**」->「**設定**」->「**チーム**」->「**無視するリソース (Ignored Resources)**」を開いて「**パターンの追加 (Add Pattern)**」クリックし、新規パターンを追加します。`resourceUpdate.sts` と入力して、「**OK**」をクリックします。無視するパターンのリストで新規パターンが選択されていることを確認し、「**OK**」をクリックして設定を保存します。

HATS Toolkit で作成された HATS Web プロジェクトは、Rational SDP ワークベンチ内の Web プロジェクトの拡張版です。詳細については、Rational SDP メニュー・バーで「**ヘルプ**」>「**ヘルプ目次**」をクリックして、「**開発 (Developing)**」を展開し、「**Web アプリケーションの開発 (Developing Web applications)**」を選択してください。

HATS Web アプリケーションの編成

デフォルトでは、すべての HATS Web アプリケーションが 1 つのエンタープライズ・アーカイブ・ファイル (HATS_EAR8 など) に保管されます。アプリケーションをエクスポートして WebSphere Application Server に配置すると、HATS .ear ファイルには、Web アーカイブ (.war) ファイルと各アプリケーションの実行に必要なリソース、および HATS ランタイム実行可能コードの 1 つのコピーが格納されます。必要があれば、各アプリケーションをそれぞれ専用の .ear ファイルに入れたり、いくつかを組み合わせて 1 つの .ear ファイルにしたりするなど、別の編成方式も使用できます。

Web アプリケーションの配置方法を決定する際には、以下の事項がサーバーに与える影響を考慮してください。

ディスク・スペース

各アプリケーションをそれぞれ専用の .ear ファイル内に作成する場合は、それぞれに専用の HATS ランタイム・コードのコピーがあります。ランタイム・コードのサイズは約 25 MB です。ランタイム・システムでご使用のアプリケーションのすべてに消費されるディスク・スペースの量を確認する必要がある場合は、25 にアプリケーションの数を掛けます。

デプロイメント

HATS .ear ファイルを再展開すると、その .ear ファイル内で一部のアプリケーションが未変更であっても、.ear ファイル内のすべてのアプリケーションが再展開されます。

ロギングとトレース

ロギングとトレースは、個々の HATS アプリケーション・レベルではなく、.ear ファイルのレベルで制御されます。各 HATS アプリケーションがそれぞれ専用の .ear ファイルに格納されている場合は、個々のアプリケーションのログとトレースの設定を、他のアプリケーションから切り離して制御することができます。 1 つの .ear ファイルに複数の HATS アプリケーションが格納されている場合は、ログとトレースの設定は、その .ear ファイル内のすべての HATS アプリケーションに適用されます。 1 つの .ear ファイル内のすべての HATS アプリケーションに関するメッセージは、同じログ・ファイルに記録され、またすべての HATS アプリケーションに関するトレース情報が同じトレース・ファイルに記録されます。

オプションのキーワード、traceLogDirectory を runtime.properties ファイルに追加できます。この操作により、出力ファイルに対して特定のディレクトリーを指定できます。このファイルは HATS Enterprise Application の *was_dir*/installedApps/*ear_name* ディレクトリーに格納されています。

ライセンス追跡

ライセンス追跡も、個々の HATS アプリケーション・レベルではなく、.ear ファイルのレベルで制御されます。各 HATS アプリケーションがそれぞれ専用の .ear ファイルに格納されている場合は、ライセンス追跡は他のアプリケーションから切り離して行われます。1 つの .ear ファイルに複数の HATS アプリケーションが入っている場合は、その .ear ファイル内のすべての HATS アプリケーションについてライセンス追跡が行われます。同じ .ear ファイル内のすべての HATS アプリケーションについてライセンス使用に関する情報が保持され、同じライセンス使用状況ファイルに入れられます。

HATS Web プロジェクトを別の .ear ファイルに移動させる

プロジェクト・ファイルがどの .ear ファイルに入るかは、プロジェクトの作成時に決定されます。プロジェクトを作成した後、そのプロジェクトをある .ear から別の .ear に移すには、以下の手順を実行します。

プロジェクトを .ear ファイルに追加するには、以下の手順を実行します。

1. HATS Toolkit の「**ナビゲーター**」タブをクリックして、.ear ファイルを表示します。
2. プロジェクトの追加先とする .ear ファイルを展開します。**META-INF** フォルダーを展開し、application.xml ファイルを見つけます。
3. application.xml ファイルをダブルクリックして、Rational SDP application.xml エディターを始動します。
4. application.xml エディターの「設計」ビューの「概要」グループで、「**アプリケーション**」を選択し、「**追加**」をクリックします。

5. 「項目の追加 (Add Item)」ダイアログで、「**モジュール**」を選択し、「**OK**」をクリックします。
6. .ear ファイルに追加するプロジェクトを選択し、「**終了**」をクリックします。
7. Rational SDP application.xml エディターを閉じます。

プロジェクトを .ear ファイルから除去するには、以下の手順を実行します。

1. HATS Toolkit の「**ナビゲーター**」タブをクリックして、.ear ファイルを表示します。
2. プロジェクトを除去する .ear ファイルを展開します。 **META-INF** フォルダーを展開し、application.xml ファイルを見つけます。
3. application.xml ファイルをダブルクリックして、Rational SDP application.xml エディターを始動します。
4. application.xml エディターの「設計」ビューの「概要」グループで、.ear ファイルから除去するプロジェクトを選択し、「**除去**」をクリックします。
5. Rational SDP application.xml エディターを閉じます。

HATS プロジェクトのバックアップ

ソフトウェア開発活動の場合と同じく、データの損失や破損に備えて HATS プロジェクトのバックアップをとっておくとよいでしょう。開発サイクル中にバックアップを作成するほかに、HATS 保守をインストールする前にもバックアップを作成する必要があります。これにより、必要な場合は、以前にインストールした保守レベルに戻すことができます。HATS プロジェクトのバックアップを作成するには、Rational SDP のアーカイブ・ファイルのエクスポート機能を使用します。詳しくは、『Web プロジェクトのエクスポート』を参照してください。

バックアップから HATS プロジェクトをリストアするには、アーカイブ・ファイルのインポート機能を使用します。詳しくは、38 ページの『Web プロジェクトのインポート』を参照してください。

HATS Web プロジェクトのエクスポートとインポート

Web プロジェクトのエクスポート

HATS Web プロジェクトをエクスポートすると、保存してバックアップとして使用したり、別の HATS Toolkit システムに移したりできます。エクスポートを実行するには、以下の方法を使用します。

アーカイブ・ファイル

この機能には、複数の特長があります。この機能では、複数のプロジェクトと関連する .ear ファイルをすべて同時にエクスポートできます。この場合、アーカイブ・ファイルをインポートする前に、宛先の HATS Toolkit でプロジェクトを作成する必要はありません。プロジェクトをアーカイブ・ファイルとしてエクスポートするには、以下の手順を実行します。

1. メニュー・バーから、「**ファイル**」>「**エクスポート**」を選択して、「エクスポート」ウィザードを開きます。
2. 「**一般**」>「**アーカイブ・ファイル**」を選択して、「**次へ**」をクリックします。

3. プロジェクトと、その関連 EAR プロジェクトの両方を選択します。
4. プロジェクトを保存するファイルのファイル名と場所を指定します。
5. 「**オプション**」セクションで、必要なファイル・フォーマットと圧縮オプションを選択します。プロジェクト全体をエクスポートするには、「**ファイルのディレクトリー構造を作成 (Create directory structure for files)**」を選択します。
6. 「**完了**」をクリックします。

Web プロジェクトのインポート

アーカイブ・ファイル (または以前のリリースの Rational SDP のプロジェクト交換ファイル) のプロジェクトを HATS Toolkit にインポートするには、以下の手順を実行します。

1. メニュー・バーから、「**ファイル**」>「**インポート**」をクリックして、「インポート」ウィザードを開きます。
2. 「**一般**」>「**既存プロジェクトをワークスペースに (Existing Projects into Workspace)**」を選択して、「**次へ**」をクリックします。
3. 「**アーカイブ・ファイルの選択 (Select archive file)**」オプションを選択し、「**参照**」をクリックしてアーカイブ・ファイルを参照します。
4. 「**プロジェクト (Projects)**」セクションで、インポートする 1 つ以上のプロジェクトを選択します。
5. 「**完了 (Finish)**」をクリックする。

インポートの後に、次のエラーが発生することがあります。

```
Project 'xxxxx' is missing required source folder: 'Java Source'

The project cannot be built until build path errors are resolved
```

このエラーを修正するには、以下の手順を実行します。

1. 「**HATS プロジェクト**」ビューでプロジェクトを選択します。
2. Rational SDP ツールバーで「**ファイル**」>「**新規**」>「**その他**」を選択します。
3. 「ウィザードの選択 (Select a wizard)」ページで「**一般**」を展開し、「**フォルダー**」を選択して「**次へ**」をクリックします。
4. 自分のプロジェクトが親フォルダーとして選択されていることを確認します。
5. フォルダー名として `Java Source` と入力します。
6. 「**完了**」をクリックします。

前のリリースの HATS から HATS プロジェクトをマイグレーションする方法について詳しくは、17 ページの『HATS のマイグレーション』を参照してください。

HATS Web アプリケーションの展開

HATS Web アプリケーション、*WebSphere アプリケーション*、および *Java EE アプリケーション* という用語は、同義で使用され、HATS プロジェクトから作成された Web アプリケーションを指します。HATS Toolkit で開発されているため、これは HATS Web アプリケーションです。また、WebSphere Application Server にインストールされて実行されるので、WebSphere アプリケーションでもあります。Java EE 標準に準拠するため、Java EE アプリケーションです。

HATS Web アプリケーションをランタイム環境に展開するには、以下を行う必要があります。

- HATS ランタイムを使用可能にします。
- プロジェクトを Java EE アプリケーションとしてエクスポートします。
- ランタイム環境でのアプリケーションのインストール
- ランタイム環境で必要な変更

以下のセクションでは、これらのタスクのそれぞれを実行する方法を説明しています。

注: HATS Web アプリケーションは、Apache Geronimo Web アプリケーション・サーバー (Tomcat Web コンテナー付属)、Oracle WebLogic Server、および IBM Bluemix Server にもデプロイできます。これらのサーバー用の HATS Web アプリケーションの開発およびデプロイに固有の情報については、以下を参照してください。

- 45 ページの『Apache Geronimo Server 用の HATS アプリケーションの開発』
- 46 ページの『Oracle WebLogic Server 用 HATS アプリケーションの開発』
- 49 ページの『IBM Bluemix Server 用の HATS アプリケーションの開発』
- 52 ページの『WebSphere Application Server Liberty Profile 用の HATS アプリケーションの開発』

HATS ランタイムの使用可能化

HATS パッケージをどこで取得するかには関係なく (HATS CD、Web、または別の製品とのパッケージ)、同じバージョンの HATS Toolkit をインストールします。これは、HATS の評価に使用できる使用制限付きバージョンです。取得済みライセンス証書に合わせて実稼働環境のランタイムを完全に使用可能にするには、HATS Toolkit に組み込まれている「ライセンス設定」ウィザードを使用して、ライセンス設定を指定する必要があります。詳しくは、「*HATS スタートアップ・ガイド*」の『HATS ランタイムおよびライセンス設定を有効にする 』を参照してください。

プロジェクトを Java EE アプリケーションとしてエクスポート

HATS Web アプリケーションを WebSphere Application Server ランタイム環境に展開するには、まず最初にこれを Java EE アプリケーションにパッケージ化する必要があります。このパッケージ化を行うには、プロジェクトを EAR ファイルとしてエクスポートします。

プロジェクトをエクスポートするには、以下のステップを実行します。

1. 「HATS Projects」ビュー内のプロジェクトの名前を強調表示 (シングルクリック) します。
2. メイン・ツールバーの「**HATS プロジェクトをエクスポート**」アイコンをクリックするか、またはプロジェクト名を右クリックして、「**プロジェクトをエクスポート**」を選択します。

   注:

ライセンス設定を指定していない場合、このプロジェクトのランタイムは使用不可であるというメッセージが表示されます。このアプリケーションを展開することはできますが、ユーザーが確立できる接続は 2 つに制限されます。

HATS Web アプリケーションはローカル・テスト環境でテストでき、また、ランタイム (非開発) 環境に展開できますが、どちらの環境でも、ライセンス設定を指定しなければ、HATS Web アプリケーションは 2 つのホスト接続しかサポートしません。ライセンス設定の指定については、*「HATS スタートアップ・ガイド」* の『HATS ランタイムおよびライセンス設定を有効にする』を参照してください。

ライセンスを購入済みの場合は、**「ランタイムの使用可能化」**をクリックして、ランタイム・アプリケーションを使用可能にします。まだ購入していない場合は、**「続行」**をクリックしてエクスポート・プロセスを続行します。

3. 「エクスポート」ウィンドウで、ドロップダウン・リストからエンタープライズ・アプリケーション・プロジェクトを選択するか、または**「EAR プロジェクト」**フィールドにエンタープライズ・アプリケーション・プロジェクトの名前を入力します。
4. **「宛先」**フィールドに宛先の場所を入力するか、または**「参照」**ボタンをクリックしてエクスポートされる .ear ファイルの宛先を選択します。
5. プロジェクトのソース・ファイルを実行可能ファイルとともにエクスポートする場合は、**「ソース・ファイルのエクスポート」**ボックスを選択します。ソース・ファイルを含めた場合は、別の開発者が .ear ファイルからこのソース・ファイルを抽出できます。これにより、コラボレーションまたはサービスがより容易になる可能性がありますが、このためにセキュリティー・リスクが生じるかどうかを判断する必要もあります。
6. 以前に、このプロジェクトを指定した場所にエクスポートしたことがある場合は、エクスポート処理で既存のファイルを上書きするかどうかが確認されます。この確認は、プロジェクトの前のバージョンを保持またはアーカイブしたいファイルを誤って上書きしないようにすることを目的としています。この確認なしで前のファイルを上書きする場合は、**「既存ファイルの上書き」**ボックスを選択してください。
7. **「完了」**をクリックします。プロジェクトが、拡張子 .ear を持つエンタープライズ・アーカイブ・ファイルによって表される Java EE アプリケーションとして、指定したディレクトリーとファイル名でエクスポートされます。

ランタイム環境でのアプリケーションのインストール

HATS プロジェクトを Java EE アプリケーションとしてエクスポートし、アプリケーションの .ear ファイルを実動システムに転送した後、WebSphere 管理コンソールを起動してそれをインストールし、.ear ファイルの場所を参照します。 WebSphere Application Server へのアプリケーションのインストールについて詳しくは、WebSphere Application Server のライブラリー (http://www.ibm.com/software/webservers/appserv/was/library/) を参照して、ご使用の WebSphere Application Server のバージョン用の Knowledge Center のリンクを選択してください。ご使用の WebSphere Application Server 製品のコンテンツで、アプリケーションの展開に関する章を参照してください。

アプリケーションがインストールされた後、別のシステム上の Web ブラウザーでこの URL を指定して開いて、これをテストします。問題がなければ、この URL をユーザーに公開することができます。URL は、例えば `http://hostname/hatsappname/` というようになります。ここで、`hostname` が WebSphere Application Server がインストールされている IP ホスト名とドメインで、`hatsappname` がご使用の HATS アプリケーションの名前です。

注:

1. 特定のプラットフォームへの HATS アプリケーションのインストールについての詳細は、HATS Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0) を参照してください。
2. HATS アプリケーションを Java 2 セキュリティーを使用可能にして WebSphere Application Server に展開した際にアプリケーションが開始しない場合、詳細については、497 ページの『Java 2 セキュリティー』を参照してください。
3. HATS は、同じ HATS アプリケーションにアクセスする同じワークステーションからの複数のブラウザー・インスタンスをサポートします。ただし、これらの複数のブラウザー・インスタンスは、同じ HTTP セッション ID を共有してはなりません。

   HATS アプリケーションにアクセスするブラウザーと WebSphere Application Server の間のセッションには、HTTP セッション ID が割り当てられます。この HTTP セッション ID によって、同じブラウザーからの後続の要求を同じ HATS アプリケーションに戻すことができます。後続の要求は、サーバー上の HATS アプリケーションとホストの間に存在する同じ Telnet (3270 または 5250) セッションを使用します。

   さまざまなブラウザーが、さまざまな方法で HTTP セッション ID を実装しています。Microsoft Internet Explorer は、同じプロセス内のすべてのブラウザー・セッションが同じ HTTP セッション ID を共用することを想定しています。同じショートカットを使用して新規ブラウザー・ウィンドウを起動すると、新規ブラウザー・セッションが開始され、同じプロセスと同じ HTTP セッション ID を共用します。

   固有の HTTP セッション ID を持つ新しい Internet Explorer プロセスを開く 1 つの方法は、例えば、Windows XP で、Internet Explorer がスタート・メニューにある場合、「**スタート**」>「**Internet Explorer**」を選択することです。「**スタート**」>「**すべてのプログラム**」>「**Internet Explorer**」を選択しても、固有の HTTP セッション ID を持つ新しい Internet Explorer プロセスが開始されます。

   注:

   a. Internet Explorer の Ctrl-N 機能では、新しいプロセスは開始されません。これは、発生元のウィンドウと同じプロセス ID と同じ HTTP セッション ID を持つ新しいウィンドウが開かれるためです。
   b. Internet Explorer 8 の場合、「**ファイル**」>「**新規セッション**」を使用して、固有のセッション ID を持つ新しいブラウザーを開くことができます。

   注:

   Internet Explorer では、メニュー・バーから「**表示**」>「**ソース**」を選択するか、または Internet Explorer ブラウザー・ウィンドウで右クリックして「**ソースの表**

示」を選択し、ページのソース・タグ設定を表示することにより、ブラウザー・ウィンドウの HTTP セッション ID を見つけることができます。ソース・ウィンドウで、次に示すような HATS Form タグまでスクロールダウンします。

```
<FORM NAME="HATSform" dir="" METHOD="POST"
ACTION="/YourHATSAppName/entry">.
```

その Form タグ内で、次に示すような SESSIONID パラメーターを見つけます。

```
<INPUT TYPE="HIDDEN" NAME="SESSIONID"
VALUE="PIOncS1GehNWxWo_49VCBzC" />
```

注: 開始時に画面のデフォルト変換の表示のみを行い、マクロは実行しない HATS アプリケーションにアクセスすると、SESSIONID の値は「INVALID」です。SESSIONID 値は、ホストに対して何かを送信 (Enter キーを押すなど) して応答を取得したときに割り当てられます。

Windows でタスク・マネージャーを使用することにより、新しいプロセスが開始されているかどうかを確認できます。タスク・マネージャーを開始し、「プロセス」タブの「イメージ名」列見出しをクリックして情報をイメージ名でソートします。スクロールダウンして、リスト内に「iexplore.exe」プロセスがいくつあるかを確認します。「**スタート**」>「**Internet Explorer**」ショートカットを使用して、新しい Internet Explorer ウィンドウをいくつか開始してみてください。新しいプロセスが開始されると、新しい iexplore.exe プロセスが表示に追加されます。それ以外の場合は、既存のプロセスが新しいブラウザー・ウィンドウを制御しています。

4. HATS では、アプリケーションの web.xml ファイルに定義されているアプリケーションの表示名が、ブラウザーの URL で使用する Web アプリケーションのコンテキスト・パスと一致している必要があります。表示名値をアプリケーションのコンテキスト・パスと異なる値に変更すると、HATS アプリケーションを実行したときに java.lang.nullPointerException エラーまたは javax.servlet.ServletException エラーが発生することがあります。例えば、MyHatsProject という名前の HATS プロジェクトでは、MyHatsProject がデフォルトの表示名です。このプロジェクトを Web アプリケーションとしてデプロイしたときに正しく作動させるには、http://myServer/MyHatsProject/ のような URL でアクセスできる必要があります。表示名を MyHatsProject 以外の何らかの値に変更すると、アプリケーションが作動不能になります。

ランタイム環境で必要な変更

クラス・ローダー・ポリシーの構成

HATS Web アプリケーションをデプロイするか、これらのアプリケーションをローカル・テスト環境で実行する場合は、以下のいずれかの状態に対して、 Java EE アプリケーションごとを基準として、WAR クラス・ローダー・ポリシーを構成する必要があります。

- アプリケーションにビジネス・ロジックが組み込まれている。このビジネス・ロジックにより、ClassNotFoundException が発生する可能性があります。
- アプリケーションにカスタム・コンポーネントまたはカスタム・ウィジェットが組み込まれている。このコンポーネントまたはウィジェットにより、ClassNotFoundException が発生する可能性があります。

HATS は、これらのアプリケーションのクラス・ローダー・ポリシーを自動的に構成します。

クラスター環境での HATS アプリケーションの構成

縦方向にクラスター化された環境に HATS Web アプリケーションを配置すると、各アプリケーション・サーバー・インスタンスは、ロギング、トレース、およびライセンス追跡用の独自のファイルを作成します。これは、出力ファイルの名前にアプリケーション・サーバー・インスタンスの完全修飾名を付け足すことによって実行されます。例えば、ロギング・ファイルのデフォルト・パターンは messages.txt ですが、実際のファイル名は messages_*myCell_myNode_myAppServerInstance*_1.txt のようになります。

デフォルトでは、すべてのサーバー・インスタンスが同じ runtime.properties ファイルの設定を読み取ります。ランタイム設定を正しく制御するには、各インスタンスが専用のランタイム・プロパティー・ファイルを持つように構成する必要があります。これにより、各インスタンスのトレースを個別に制御できるようになり、ランタイム設定が意図せず変更されるのを防ぐことができます。

縦方向にクラスター化された HATS アプリケーションのインスタンスがそれぞれ専用の runtime.properties ファイルを持つように構成するには、次の手順を実行します。

1. 縦方向のクラスター内のアプリケーション・サーバー・インスタンスごとに、runtime.properties ファイルのコピーを作成します。
   a. ご使用の HATS アプリケーションの runtime.properties ファイルを見つけます。このファイルは、WebSphere Application Server をインストールしたディレクトリーの下の installedApps¥*app_name.ear* ディレクトリーにあります。
   b. 各インスタンスで、runtime.properties ファイルのコピーを作成します。そのファイルは固有の名前が付けられ、同じディレクトリーに配置されます。例えば、ファイル Clone1runtime.properties、Clone2runtime.properties などの名前を付けます。有効なファイル名であれば何でもかまいませんが、このファイルに関連するアプリケーション・サーバー・インスタンスを管理者が特定できるような名前にしてください。
   c. この時点で、n 個のサーバー・インスタンスに対して固有のランタイム・プロパティー・ファイルを n 個保持しています。
   d. 縦方向にクラスター化された環境内に複数の HATS .ear ファイルがある場合は、各 .ear ファイルのディレクトリーでこの手順を繰り返します。
2. 新規構成設定を追加して、各インスタンスで使用する runtime.properties ファイルを識別します。例えば、WebSphere Application Server V6.x の場合は、以下のようにします。
   a. WebSphere Application Server 管理コンソールの左側のナビゲーション・ペインにある「**サーバー**」項目の「**アプリケーション・サーバー**」を選択します。
   b. アプリケーション・サーバーのリストからサーバー・インスタンスを選択します。
   c. サーバーの「**構成**」タブから、「**Java およびプロセス管理**」を選択します。

d. 「**プロセス定義**」>「**Java 仮想マシン**」>「**カスタム・プロパティー**」を選択します。
e. 「カスタム・プロパティー」ウィンドウで、「**新規**」をクリックします。
f. 「**名前**」フィールドに `hats.runtime.properties` と入力します。
g. 「**値**」フィールドで、Clone2runtime.properties など、このサーバーに対して作成したプロパティー・ファイルの名前を入力します。この値でディレクトリー名やスラッシュは指定しないでください。
h. 「**説明**」フィールドには値は必要ありません。
i. 「**適用**」をクリックします。
j. 各サーバー・インスタンスに対してこの手順を繰り返します。

新規カスタム・プロパティーを追加して、各サーバー・インスタンスに固有の名前の付いた専用の runtime.properties ファイルのコピーがあることを確認したら、新規ファイルの使用を開始するように各アプリケーションを再始動する必要があります。

HATS 管理コンソールは、クラスター・メンバーの設定を制御するのに使用できます。「管理 Web (Administration Web)」ページの「ナビゲーション (Navigation)」パネルの「**スタートアップ・ガイド**」フォルダーを使用して、「**管理の有効範囲**」を選択します。クラスターを選択した後、「**トレース設定**」の表示を選択すると、制御する特定のクラスター・メンバーの入力を要求されます。

注: クラスタリングとユーザー・リストを使用する場合は、以下のような特別な考慮事項があります。詳しくは、172 ページの『クラスタリングとユーザー・リスト』を参照してください。

プロキシー・サーバーを使用するための HATS アプリケーションの構成

ユーザーがプロキシー・サーバーを介して HATS アプリケーションにアクセスする場合、プロキシー・サーバーと、HATS アプリケーションのコンテキスト・パラメーターの両方を構成する必要があります。

プロキシー・サーバーを構成するには、以下の手順を実行します。

1. プロキシー・サーバーで、httpd.conf ファイルを見つけます。
2. メモ帳などのテキスト・エディターを使用して、以下の 2 つの行をファイルに追加します。

```
ProxyPass /application_name/ http://yyyy:port/application_name/
ProxyPassReverse /application_name/ http://yyyy:port/application_name/
```

ここで、*application_name* は HATS アプリケーションの名前、*yyyy* は、HATS アプリケーションがインストールされているアプリケーション・サーバーの完全修飾 IP アドレス、および *port* はポート番号です。

HATS アプリケーションでコンテキスト・パラメーターを構成するには、以下の手順を実行します。

1. HATS アプリケーションの web.xml ファイル (「**ナビゲーター**」ビューの Web Content¥WEB-INF フォルダーにあります) を編集します。
2. 以下のコンテキスト・パラメーターを追加します。

```
<context-param>
  <param-name>com.ibm.hats.proxyURL</param-name>
  <param-value>http://myproxyserver.com:port/application_name</param-value>
</context-param>
```

ここで、*myproxyserver.com* はプロキシー・サーバーの URL、`port` はポート番号、*application_name* は HATS アプリケーションの名前です。

Apache Geronimo Server 用の HATS アプリケーションの開発

Tomcat Web コンテナーを持つ Apache Geronimo Web アプリケーション・サーバーを対象とした HATS Web アプリケーションを開発、テスト、およびデプロイすることができます。サポートされる Apache Geronimo のリリースについては、「Rational Host Access Transformation Services: System Requirements」(http://www.ibm.com/support/docview.wss?uid=swg27011794) を参照してください。資料やダウンロードを含め、Apache Geronimo に関する詳細情報については、「Welcome to Apache Geronimo」サイト (http://geronimo.apache.org) を参照してください。

Geronimo サーバー用のインストールおよび構成

このセクションでは、Apache Geronimo をインストールする方法と、Apache Geronimo サーバーを対象とした HATS Web アプリケーションを作成できるように Rational Software Delivery Platform (Rational SDP) を構成する方法の要約を示します。

Apache Geronimo サーバーを Tomcat Web コンテナーとともにダウンロードおよびインストールします。

1. Apache Geronimo をダウンロード・サイト (http://geronimo.apache.org/downloads.html) からダウンロードします。
2. .zip ファイルまたは .tar ファイルを解凍して、サーバーをインストールします。
3. コマンド・ライン・ウィンドウで、ディレクトリーを *geronimo_home*/bin に変更し、`geronimo run` を入力して、サーバーを実行します。

Geronimo Eclipse Plugin (GEP) は、Eclipse、その Web Tools Platform (WTP) プロジェクト、および Geronimo の間の統合を提供します。これにより、WTP フィーチャーを利用して、Rational SDP を使用した Apache Geronimo サーバー用アプリケーションの開発、テスト、およびデプロイを行うことが可能になります。

Geronimo Eclipse Plugin (GEP) をインストールします。

1. 「Installing Geronimo Eclipse Plugin」サイト (https://cwiki.apache.org/GMOxDOC22/installing-geronimo-eclipse-plugin.html) の指示に従います。
2. GEP v2.2 には、v1.1.x、v2.0.x、v2.1.x、および v2.2 の Apache Geronimo サーバー用のサーバー・アダプターが含まれています。

Geronimo を Rational SDP で、HATS Web アプリケーション用にターゲット・サーバーとして構成します。

1. Rational SDP メニュー・バーで「**ウィンドウ**」>「**設定**」をクリックします。
2. 左のパネルで「**サーバー**」を展開します。
3. 「**ランタイム環境**」をクリックします。
4. 「サーバー・ランタイム環境」パネルで「**追加**」をクリックします。

5. 「新規サーバー・ランタイム環境」ダイアログで「**Apache**」フォルダーを展開します。
6. **Apache Geronimo** サーバーの当該バージョンを選択します。
7. オプション。「**新規ローカル・サーバーの作成 (Create a new local server)**」ボックスを選択して、新しい Geronimo サーバーをワークスペースの「**サーバー**」タブで作成します。このステップを行うと、「サーバーで実行」、「サーバーでデバッグ」、および「サーバーでプロファイル作成」の各機能を使用して、Rational SDP ローカル・テスト環境内で Geronimo アプリケーションをテストすることができます。
8. 「**次へ**」をクリックして、ローカル Geronimo サーバーを指示するパネルに移動します。「アプリケーション・サーバー・**インストール・ディレクトリー**」フィールドで、Geronimo サーバー・ファイルを解凍したルート・フォルダーを入力します。例えば、Geronimo サーバー zip を `C:¥geronimo-tomcat6-javaee5-2.1.7` というフォルダーに解凍した場合は、この値をインストール・ディレクトリー・フィールドに入力します。
9. 「**完了**」をクリックします。
10. 「**OK**」をクリックして「設定」パネルを閉じます。

これらの手順を終えると、Rational SDP 内で Apache Geronimo サーバーを対象とした HATS アプリケーションを作成することができます。

Geronimo サーバーに関する考慮事項および制約事項

Apache Geronimo サーバーと WebSphere Application Server には、HATS Web アプリケーションの開発、テスト、および実行に関して、異なる考慮事項がいくつかあります。例えば、Apache Geronimo サーバーで実行されている .ear ファイルごとにサポートされる HATS Web アプリケーションは 1 つのみです。

最新のサポートに関する考慮事項については、「Host Access Transformation Services 9.5 - Known issues and workarounds」(http://www.ibm.com/support/docview.wss?uid=swg27046998) を参照してください。

Oracle WebLogic Server 用 HATS アプリケーションの開発

Oracle WebLogic Server を対象とした HATS Web アプリケーションを開発、テスト、およびデプロイできます。サポートされる Oracle WebLogic Server のリリースについては、「Rational Host Access Transformation Services: System Requirements」(http://www.ibm.com/support/docview.wss?uid=swg27011794) を参照してください。資料やダウンロードを含め、Oracle WebLogic Server について詳しくは、Oracle Web サイト (http://www.oracle.com/technetwork/middleware/weblogic/overview/index.html) を参照してください。

WebLogic サーバーのインストールおよび構成

このセクションでは、Oracle WebLogic Server をインストールする方法と、Oracle WebLogic サーバーを対象とした HATS Web アプリケーションを作成できるように Rational Software Delivery Platform (Rational SDP) を構成する方法の要約を示します。

Oracle サイト (http://www.oracle.com/technetwork/middleware/weblogic/downloads/index.html) のダウンロード・リンクおよび指示を使用して Oracle WebLogic Server をダウンロードおよびインストールします。

Oracle WebLogic Server Tools を 9.0 より前のバージョンの RAD にインストールします。

1. Rational SDP メニュー・バーで「**ヘルプ**」>「**新規ソフトウェアのインストール (Install New Software)**」をクリックします。
2. 「**ランタイム環境**」をクリックします。
3. 「サーバー・ランタイム環境」パネルで「**追加**」をクリックします。
4. 「**追加サーバー・アダプターのダウンロード (Download additional server adapters)**」をクリックします。
5. 「**Oracle WebLogic Server ツール (Oracle WebLogic Server Tools)**」をリストから選択し、「**次へ**」をクリックします。
6. ご使用条件の条項に同意し、「**完了**」をクリックします。
7. インストールの完了後に再始動を求めるプロンプトが出されたら、「**OK**」をクリックします。
8. インストールの完了後、今すぐ再始動するように求めるプロンプトが出されたら、「**はい**」をクリックします。

Oracle WebLogic Server Tools を RAD 9.0 にインストールします。

1. Rational SDP メニュー・バーで「**ウィンドウ**」>「**設定**」をクリックします。
2. 右上のパネルで「**使用可能なソフトウェア・サイト (Available Software Sites)**」をクリックします。
3. 「**追加**」をクリックします。
4. 以下の情報を入力して必要なリポジトリーを追加します。

   名前: Oracle Enterprise Pack for Eclipse Repository

   場所: http://download.oracle.com/otn_software/oepe/12.1.2/juno/repository
5. 「**OK**」をクリックします。
6. 「使用可能なソフトウェア・サイト (Available Software Sites)」リストにある以下の追加リポジトリーを使用可能にするか、またはこれらのリポジトリーがない場合は追加します。

   名前: Eclipse Juno repository

   場所: http://download.eclipse.org/releases/juno

   名前: Eclipse Web Tools Platform Repository

   場所: http://download.eclipse.org/webtools/updates/
7. 「**OK**」をクリックして「設定」ウィンドウを閉じます。
8. 「**必要なソフトウェアを検索するために、インストール時にすべての更新サイトに接続する (Contact all update sites during install to find required software)**」チェック・ボックスを選択します。

9. 「**処理 (Work With)**」>「**Oracle Enterprise Pack for Eclipse Repository**」を選択します。
10. 結果を展開し、「**Oracle WebLogic Server Tools**」チェック・ボックスを選択します。
11. 「**次へ**」をクリックします。
12. ウィザードの指示に従ってインストールを完了します。

Oracle WebLogic Server Tools を RAD 9.5 にインストールします。

1. Rational SDP メニュー・バーで「**ヘルプ**」>「**Eclipse Marketplace**」をクリックします。
2. 「検索」テキスト・ボックスで「**WebLogic**」を検索します。
3. Oracle WebLogic Server Tools をインストールします。
4. 「選択したフィーチャーの確認 (Confirm Selected Features)」パネルで「**確認**」をクリックします。
5. ご使用条件の条項を読んで条項に同意し、「**完了**」をクリックします。
6. インストールが完了した後、再始動するかどうかを尋ねるメッセージが表示されたら、「**はい**」をクリックします。

WebLogic を Rational SDP で、HATS Web アプリケーション用にターゲット・サーバーとして構成します。

1. Rational SDP メニュー・バーで「**ウィンドウ**」>「**設定**」をクリックします。
2. 左のパネルで「**サーバー**」を展開します。
3. 「**ランタイム環境**」をクリックします。
4. 「サーバー・ランタイム環境」パネルで「**追加**」をクリックします。
5. 「新規サーバー・ランタイム環境」ダイアログで「**Oracle**」フォルダーを展開します。
6. **Oracle WebLogic Server** の当該バージョンを選択します。
7. オプション。「**新規ローカル・サーバーの作成 (Create a new local server)**」ボックスを選択して、新しい WebLogic サーバーをワークスペースの「**サーバー**」タブで作成します。このステップを実行すると、「サーバーで実行」、「サーバーでデバッグ」、および「サーバーでプロファイル作成」の各機能を使用して、Rational SDP ローカル・テスト環境内で WebLogic アプリケーションをテストすることができます。

   注: このステップを実行する場合は、「サーバーで実行」、「サーバーでデバッグ」、および「サーバーでプロファイル作成」の各機能の使用時にプロジェクトが正しくデプロイされるように、デフォルトの公開モードを変更する必要があります。

   a. 「**サーバー**」ビューで作成された WebLogic Server のインスタンスを見つけます。
   b. WebLogic Server を右クリックし、「**プロパティー**」をクリックします。
   c. 左のパネルで **WebLogic** を展開します。
   d. 「**公開 (Publishing)**」をクリックします。

e. 「公開モード (Publishing mode)」セクションで、**「展開されたアーカイブとして公開 (Publish as an exploded archive)」**をクリックします。

f. **「OK」**をクリックして「プロパティー」パネルを閉じます。

8. **「次へ」**をクリックして、ローカル WebLogic サーバーを指示するパネルに移動します。**「WebLogic ホーム」**フィールドに、WebLogic サーバーをインストールしたルート・フォルダーを入力します。例えば、WebLogic インストール済み環境で `C:¥Oracle¥Middleware` というルート・フォルダーを使用している場合は、**「WebLogic ホーム」**フィールドに `C:¥Oracle¥Middleware¥wlserver_12.1` と入力します。 WebLogic サーバー・インストーラーに組み込まれている JDK をインストールした場合は、`C:¥Oracle¥Middleware¥jdk160_29` などの値が自動的に**「JAVA ホーム」**フィールドに入れられます。組み込み JDK をインストールしていない場合は、**「JAVA ホーム」**フィールドに、使用している Java ランタイムの場所を入力する必要があります。
9. **「完了」**をクリックします。
10. **「OK」**をクリックして「設定」パネルを閉じます。

これらの手順を終えると、Oracle WebLogic Server を対象とした HATS アプリケーションを Rational SDP 内で作成することができます。

WebLogic サーバーに関する考慮事項および制約事項

Oracle WebLogic Server と WebSphere Application Server には、HATS Web アプリケーションの開発、テスト、および実行に関して、異なる考慮事項がいくつかあります。例えば、Oracle WebLogic Server で実行されている .ear ファイルごとにサポートされる HATS Web アプリケーションは 1 つのみです。

最新のサポートに関する考慮事項については、「Host Access Transformation Services 9.5 - Known issues and workarounds」(http://www.ibm.com/support/docview.wss?uid=swg27046998) を参照してください。

IBM Bluemix Server 用の HATS アプリケーションの開発

IBM Bluemix Server を対象とした HATS Web アプリケーションを開発、テスト、およびデプロイすることができます。資料など、IBM Bluemix Server について詳しくは、Bluemix Web サイト (http://www.ibm.com/cloud-computing/bluemix/) を参照してください。 Rational SDP Bluemix フィーチャーについて詳しくは、http://www.ibm.com/support/docview.wss?uid=swg27046332 を参照してください。

IBM Bluemix サーバー用の Rational SDP の構成

このセクションでは、IBM Bluemix サーバーを対象とした HATS Web アプリケーションを作成できるように Rational Software Delivery Platform (Rational SDP) を構成する方法について説明します。

IBM Bluemix サーバーを HATS Web アプリケーション用に Rational SDP でターゲット・サーバーとして構成します。

1. Rational SDP メニュー・バーで**「ウィンドウ」>「設定」**をクリックします。
2. 左のパネルで**「サーバー」**を展開します。

3. 「**ランタイム環境**」をクリックします。
4. 「サーバー・ランタイム環境」パネルで「**追加**」をクリックします。
5. 「新規サーバー・ランタイム環境 (New Server Runtime Environment)」ダイアログで「**IBM**」フォルダーを展開します。
6. 「**IBM Bluemix ランタイム (IBM Bluemix Runtime)**」を選択します。
7. オプション: 「**新規ローカル・サーバーの作成 (Create a new local server)**」ボックスを選択して、新しい Bluemix サーバーをワークスペースの「サーバー (Servers)」タブで作成します。このステップを行うと、「サーバーで実行」、「サーバーでデバッグ」、および「サーバーでプロファイル作成」の各機能を使用して、Rational SDP ローカル・テスト環境内で Bluemix アプリケーションをテストすることができます。
8. 「**次へ**」をクリックして、Bluemix アカウントの詳細を入力するパネルに移動します。Bluemix アカウントの E メール・アドレスとパスワードを入力し、アカウントを検証します。
9. 「**次へ**」をクリックして、組織とスペースを取得するパネルに移動します。必要なスペースを選択します。
10. 「**完了**」をクリックしてから、「**OK**」をクリックして「設定」パネルを閉じます。

これらの手順を終えると、IBM Bluemix Server を対象とした HATS アプリケーションを Rational SDP 内で作成することができます。

IBM Bluemix Server に関する考慮事項および制約事項

IBM Bluemix Server と WebSphere Application Server には、HATS Web アプリケーションの開発、テスト、および実行に関して、異なる考慮事項がいくつかあります。

- IBM Bluemix Server で実行されている .ear ファイルごとに、1 つの HATS Web アプリケーションがサポートされます。
- HATS 管理コンソールはサポートされていません。
- 「端末の表示」はサポートされていません (trace.HOD.DISPLAYTERMINAL)。
- IBM Bluemix は、Rational SDP を使用する IBM WebSphere JAX-WS Web サービス・ランタイムをサポートしていません。
- HATS トレース/ログ・ファイルの表示方法

Bluemix サーバー内のログ・ファイルの場所:

```
/home/vcap/app/wlp/usr/servers/defaultServer/apps/myapp.ear/logs/
```

次に示すように、基本的に Bluemix サーバー・コンソール (HATS Studio コンソール) でトレース・ファイルのパスを監視することができます。

SystemOut O HAT0002
メッセージ・ログ・ファイルは、次のとおりです。

```
/home/vcap/app/wlp/usr/servers/defaultServer/apps/
      myapp.ear/logs/messages_XXXXXXXX_XX.txt
```

Bluemix Web サイトを使用してログ・ファイルを表示するには、次のようにします。

- ブラウザーで Bluemix ホーム・ページにログインします。
- HATS アプリケーションの「**概要**」ページを開きます。
- 左側のメニューで「**ログおよびファイル (Files and Logs)**」をクリックします。
- 上で示されたパスにナビゲートします。パスの先頭は以下のとおりです。

  ```
  /app/wlp/usr/servers/defaultServer/apps/myapp.ear/logs/
  ```

Rational SDP の「リモート・システム (Remote Systems)」ビューを使用してログ・ファイルを表示するには、次のようにします。

- Rational SDP の「サーバー (Servers)」ビューで「**IBM Bluemix Server**」をダブルクリックします。
- デフォルトでは、「**概要**」タブが開きます。「**アプリケーションおよびサービス (Applications and Services)**」タブに移動します。
- 「**アプリケーション**」の下で、ご使用の HATS アプリケーションを選択します。
- 右下にある「**リモート・システム (Remote Systems)」ビュー**のリンクをクリックします。Bluemix Server 用に「リモート・システム (Remote Systems)」ビューが開きます。ご使用の HATS アプリケーションが表示されるまでナビゲートします。
- 上で示された、ログ・ファイルのパスにナビゲートします。

注:

1. Bluemix ランタイムを Rational SDP に追加しているときに、次のエラーが表示されることがあります。

   ```
   java.lang.ClassNotFoundException:
   指定されたクラス com.ibm.websphere.ssl.protocol.SSLSocketFactory
   が見つかりません
   (Cannot find the  specified class
   com.ibm.websphere.ssl.protocol.SSLSocketFactory)
   ```

   SSLSocketFactory エラー・メッセージに対する回避策としては、`x:¥workspace¥.metadata¥.plugins¥org.eclipse.core.runtime¥.settings` (`x:¥workspace` はワークスペースのディレクトリー) で入手可能な `com.ibm.ws.ast.st.core.prefs` ファイルを見つけて、`isUseIBMSSLSocketFactory=true` の値を 「`false`」に置き換えます。その後、ワークベンチを再始動します。

2. Bluemix Server に HATS ear をデプロイしているときに、Bluemix コンソールで次の例外が発生します。

   ```
   FFDC1015I: FFDC インシデントが作成されました
   (An FFDC Incident has been created):
   "com.ibm.wsspi.adaptable.module.UnableToAdaptException:
   com.ibm.ws.javaee.ddmodel.DDParser$ParseException:
   CWWKC2262E: サーバーはバージョン 7、および /META-INF/application.xml
   デプロイメント記述子の 2 行目にある http://xmlns.jcp.org/xml/ns/javaee
   名前空間を処理できません。
   (The server is unable to process the 7 version and the
   http://xmlns.jcp.org/xml/ns/javaee namespace in the
   /META-INF/application.xml deployment descriptor on line 2.)
   com.ibm.ws.app.manager.ear.internal.EARDeployedAppInfoFactoryImpl 132"
   場所: ffdc_15.09.11_05.03.09.0.lo .....
   (at ffdc_15.09.11_05.03.09.0.lo .....)
   ```

デフォルトでは、Bluemix サーバー用の HATS アプリケーションは J2EE 7 バージョンを使用して作成されます。そのため、Java EE 7 機能セットのうちの主要な部分を組み込むために、ご使用の HATS Bluemix アプリケーションに次の環境変数を設定することができます。その後で、追加機能を必要に応じて追加するか削除することができます。HATS アプリケーションを Bluemix サーバーにデプロイするときに、この環境変数を構成することができます。

```
名前: JBP_CONFIG_LIBERTY
値: app_archive: {features: [ webProfile-7.0, cdi-1.2, jaxrs-2.0,
jpa-2.1, websocket-1.1, servlet-3.1, jsp-2.3, ejbLite-3.2 ]}
```

詳しくは、記事 https://developer.ibm.com/bluemix/2015/08/28/updates-to-ibm-eclipse-tools-for-bluemix-august-2015/ を参照してください。

Bluemix サーバーを使用して HATS ビジネス・ロジックまたはカスタム Java ファイルのリモート・デバッグを行うには、http://www.ibm.com/support/docview.wss?uid=swg27046332#whatsnew_95_bluemix_debug を参照してください。

WebSphere Application Server Liberty Profile 用の HATS アプリケーションの開発

以下のステップを実行して、WebSphere Application Server Liberty Profile を対象とした HATS アプリケーションを Rational SDP 内で作成できます。

1. Rational SDP メニュー・バーで、「**ウィンドウ**」 > 「**設定**」をクリックします。
2. 左のパネルで「**サーバー**」を展開します。
3. 「**ランタイム環境**」をクリックします。
4. 「サーバー・ランタイム環境」パネルで「**追加**」をクリックします。
5. 「新規サーバー・ランタイム環境 (New Server Runtime Environment)」ダイアログで「**IBM**」フォルダーを展開します。
6. 「**WebSphere Application Server Liberty Profile**」を選択します。
7. オプション: 「サーバーで実行」、「サーバーでデバッグ」、および「サーバーでプロファイル作成」の各機能を使用して、Rational SDP ローカル・テスト環境内で Liberty Profile アプリケーションをテストすることができます。「**新規ローカル・サーバーの作成 (Create a new local server)**」ボックスを選択して、新しい Liberty Profile サーバーをワークスペースの「**サーバー (Servers)**」タブで作成します。
8. 「**次へ**」をクリックして、ローカル Liberty Profile サーバーを指定するパネルに移動します。「**既存のインストール済み環境の選択 (Choose an existing installation)**」ラジオ・ボタンをクリックして、WebSphere Application Server Liberty Profile がインストールされている場所を「**パス**」フィールドに入力します。

   注: Liberty Profile サーバーをインストールしていない場合は、「**新規サーバー・ランタイム環境 (New Server Runtime Environment)**」ダイアログ・ボックスから直接インストールできます。以下にそのステップを示します。

   a. 「**アーカイブまたはリポジトリーからインストール (Install from an archive or repository)**」ラジオ・ボタンをクリックします。

b. 「**次へ**」をクリックして、「ランタイム環境のインストール (Install Runtime Environment)」パネルに進みます。

c. C:¥Liberty などのパスを「**宛先パスを入力してください (Enter the destination path)**」フィールドに入力します。

d. 「**新規ランタイム環境を ibm.com からダウンロードしてインストール (Download and install a new runtime environment from ibm.com)**」ラジオ・ボタンをクリックして、ドロップダウン・メニューで「**IBM WebSphere Liberty Repository**」を選択します。

e. WAS Liberty V8.5.5.7 ランタイムなどのランタイム・フィーチャーが含まれている WAS Liberty サーバーの 1 つをリストから選択し、「**次へ**」をクリックします。

f. 追加のバンドルが必要な場合は「追加コンテンツのインストール (Install Additional Content)」パネルで選択し、「**次へ**」をクリックします。

g. ご使用条件の条項に同意し、「**完了**」をクリックします。

9. 「**完了**」をクリックします。
10. 「**OK**」をクリックして「設定」パネルを閉じます。

WebSphere Application Server Liberty Profile に関する考慮事項および制約事項

「サーバーでデバッグ」を使用したアプリケーションのテスト時に、端末ウィンドウで表示の問題が発生する可能性があります。次のエラー・メッセージが表示される場合があります。

```
エラー HPS5018 予期しない例外が発生しました。
フレームを表示できません: java.awt.HeadlessException。
(Frame unable to display: java.awt.HeadlessException.)
```

このエラーがコンソールで発生した場合は、以下の手順に従ってこの問題を修正できます。

1. 「HATS プロジェクト」ビューで「**サーバー (Servers)**」タブを開きます。
2. リストされた対象の WebSphere Application Server Liberty Profile サーバーを右クリックして、「**新規**」 > 「**サーバー環境ファイル (Server Environment File)**」 > 「**jvm.options**」をクリックします。
3. 次の値をファイルに追加します。

   ```
   -Djava.awt.headless=false
   ```

4. jvm.options ファイルを保存して閉じます。
5. 変更内容を有効にするためにサーバーを再始動します。

モバイル装置用の HATS アプリケーションの作成

携帯電話、データ収集端末、および携帯情報端末 (PDA) などのモバイル装置からアクセス可能な HATS Web アプリケーションを開発することができます。開発プロセスは、HATS Web アプリケーションを開発する場合と同じですが、画面サイズや対話の制限など、モバイル装置とその Web ブラウザーの機能に関するいくつかの考慮事項があります。

モバイル装置用のアプリケーションのプロジェクトを作成するには、次のいずれかの操作によって「プロジェクトの作成」ウィザードを起動します。

- メニュー・バーから「**HATS**」>「**新規**」>「**プロジェクト**」を選択します。
- メニュー・バーから「**ファイル**」>「**新規**」>「**HATS プロジェクト**」を選択します。
- ツールバーの「**HATS プロジェクトを作成**」アイコンをクリックします。

「プロジェクトの作成」ウィザードの「HATS プロジェクト」パネルで、以下のようにします。

1. モバイル装置プロジェクトの名前を入力します。
2. オプションで、説明を入力します。
3. デフォルトの場所をそのまま使用します。
4. 「**Web**」を選択して、これが Web プロジェクトであることを示します。

   注: Web デプロイメント・オプションが使用不可になっている場合、サーバー・ランタイムが定義されていないことを示しています。サーバー・ランタイムを定義するには、「**ウィンドウ**」>「**設定**」>「**サーバー**」>「**インストール済みランタイム**」と移動して、少なくとも 1 つのランタイム定義を追加します。
5. 「ターゲット・サーバー」フィールドで、サポートされるいずれかの WebSphere Application Server を選択します。
6. 「エンタープライズ・アプリケーション・プロジェクト」フィールドに、ローカル・テスト環境でのテストに使用する EAR プロジェクトの名前を入力します。
7. 「**モバイル装置用にオプションを最適化**」を選択します。

   注: iPad デバイス用の HATS アプリケーションを開発する場合は、このオプションを選択しないでください。詳しくは、62 ページの『iPad デバイスに関する考慮事項および制約事項』を参照してください。
8. 「**次へ**」をクリックして、プロジェクトの作成を通常どおりに続行します。

モバイル装置に関する考慮事項および制約事項

ユーザーが HATS プロジェクトを作成し、「**モバイル装置用にオプションを最適化**」を選択すると、HATS は、モバイル装置に最も適したオプションを使用してプロジェクトを初期化します。印刷、キーボード、非同期更新などの一部のオプションはサポートされていないため、使用不可に設定されています。モバイル装置用のアプリケーションを開発する場合は、次の考慮事項と制約事項に注意してください。

- モバイル装置用に最適化されていない HATS プロジェクトを、モバイル装置用に最適化された HATS プロジェクトに変換 (またはその反対) できるようにする自動化されたオプションはありません。
- プロジェクトのテーマを指定するためのオプションは提供されていません。代わりに、モバイル装置で最適に機能するようにオプションが自動的に初期化されます。
- プロジェクトで使用するために、モバイル装置用に最適化されたテンプレートのみ提供されています。

注: Modern テンプレートを使用するときに水平スクロールが不要な場合は、Modern.jsp テンプレート・ファイルの `.roundedcornermodule` に `overflow: auto;` スタイルを追加してください。例えば、`.roundedcornermodule {margin: 0px;}` を `.roundedcornermodule { margin: 0px; overflow: auto; }` に変更します。これにより、ページが元のサイズを保持するよう強制され、装置上の可視画面内に入るよう強制されます。この結果、場合によっては、ページ上のデータが切り捨てられることがあります。したがって、カスタマイズされたページとウィジェット選択の設計では、画面サイズを考慮する必要があることに注意してください。

- 「**コンパクト (compact)**」 という第 2 レンダリング・セットがプロジェクトで作成されます。このレンダリング・セットは、デフォルト・レンダリング・セットとして指定されます。また、「**コンパクト・レンダリングの使用**」オプションが選択されており、デフォルト・レンダリングでの HTML と空白スペースの量が削減されるため、順に元のホスト画面とは異なる構造が表示される場合があります。
- HATS 設定の「**フリー・レイアウト・テーブルを含める**」は、新規ブランク変換を作成する際に有効になりますが、追加の修飾子として、「**プロジェクトがモバイル装置用に最適化されている場合を除く**」を持っています。この修飾子はデフォルトで選択されています。そのため、モバイル・プロジェクトに追加されたブランク変換には、デフォルトではフリー・レイアウト・テーブルは含まれません。
- 「フィールドの終了」、「フィールド+」、および「フィールド-」は、「**入力フィールドのカーソル位置オプションを使用可能にする**」オプションを「ホスト・キーパッド」または「ホスト・キー」ボタンと組み合わせることで使用できます。
- フィールド・ウィジェットには、テーブルではなくインライン・スパン・タグを使用して出力をレンダリングし、フィールド間の識別を行うための、「**分離**」レイアウト・オプションが用意されています。このオプションを使用する目的は、HTML および空白スペースの量を減らすことです。これはモバイル・プロジェクトのデフォルト設定です。
- HATS には、サブファイル・ウィジェットおよびテーブル・ウィジェット用の「**列の配置**」サポートが用意されています。このサポートは、モバイル装置にテーブル・データを表示する際に有用であり、ディスプレイで列の配置や除外を行ったり、拡張可能な詳細セクションを使用して、テーブルをより小さなスペースに収めたりすることができます。

  詳細セクションは、拡張時にデータの基本列を含む行のすぐ下に表示されます。該当する特定の行がユーザーにより識別されると、その行の詳細は、拡張されたときに画面の右方向ではなく下方向に続く形式で表示されます。これにより、小さな画面でも、水平スクロールを使用せずに、任意の数の列からなる必要な詳細を表示できるようになります。

  以下の図は、ホスト画面上に表示されるサブファイルを示しており、その次の図はそれぞれ、デフォルトの「列の配置」サポートを使用したときのサブファイルの基本ビューおよびサブファイルの詳細ビューを示しています。

```
 1= Add license key  2=Change  5=Display detail  6=Print detail
 8=Work with license users ...

               License
Opt   Product    Term     Feature  Description
__    5722SS1   V5R4M0     5050    i5/OS
__    5722SS1   V5         5051    i5/OS
__    5722SS1   V5R4M0     5103    Media and Storage Extensions
__    5722SS1   V5         5109    NetWare Enhanced Integration
__    5722SS1   V5R4M0     5112    PSF 1-45 IPM Printer Support
__    5722SS1   V5R4M0     5113    PSF 1-100 IPM Printer Support
__    5722SS1   V5R4M0     5114    PSF Any Speed Printer Support
                                                              More...
```

図 3. ホスト画面上のサブファイル

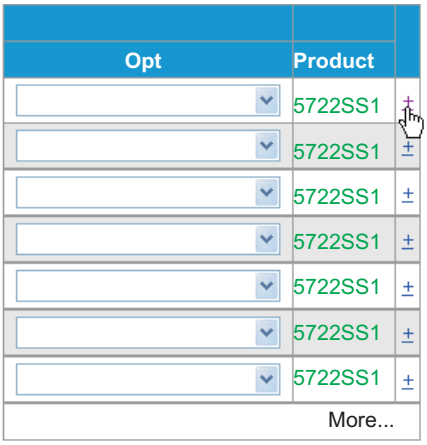


図 4. デフォルトの「列の配置」サポートを使用したときのサブファイルの基本ビュー

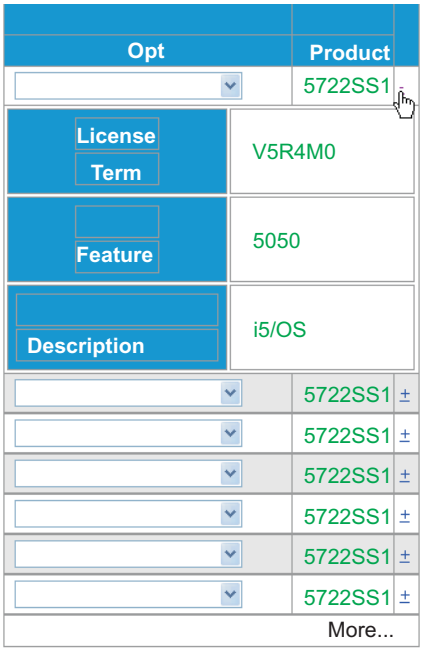


図 5. デフォルトの「列の配置」サポートを使用したときのサブファイル行の詳細ビュー

画面のサイズの保存に加えて、詳細の列を、HTTP を使用してユーザーのブラウザーに送信せずに、要求されるまでサーバー上に保持しておくようウィジェットを構成できます。これにより、不要な詳細データが HTTP を介して終端装置に送られることがなくなるため、転送されるデータの量を削減できる場合があります。ユーザーによって具体的に要求された詳細のみが、オンデマンドで取り出されて、ブラウザーに送信されます。

注: サブファイルのデータ部分を認識するためのフィールド・コンポーネントを使用するよう構成されたサブファイルは、基本列機能および詳細列機能を有効に使用することはできません。これは、このようなサブファイルは行しか識別できず、列は識別できないからです。

258 ページの『ウィジェット設定』で説明されている「**列の配置**」サブファイルおよびテーブル・ウィジェットの設定に加えて、次の設定を使用して、このオプションで使用するコントロールの外観をさらにカスタマイズすることができます。これらの設定は、HATS Toolkit GUI には表示されません。代わりに、ウィジェットのソースにこれらの設定を追加する必要があります。以下のソースは、例としてサブファイル・ウィジェットを使用したときのこれらの設定のサンプルです。

```
<class name="com.ibm.hats.transform.widgets.SubfileWidgetV6">
     <setting name="normalColumnLayout" value="1,2"/>
     <setting name="extendedColumnLayout" value="3*"/>
     <setting name="keepExpansionOnServer" value="true"/>
     <setting name="expandHeaderValue" value="fred"/>
     <setting name="expandRepresentation" value="button"/>
     <setting name="expandValue" value="click me"/>
     <setting name="collapseRepresentation" value="image"/>
     <setting name="collapseValue" value="/common/images/twisty1.gif"/>
     <setting name="collapseAltValue" value="hide"/>
</class>
```

expandRepresentation

詳細を表示するコントロールに使用するグラフィカル表現を決定します。「**button**」、「**link**」、または「**image**」を指定します。デフォルトは「**link**」です。

expandValue

expandRepresentation が「**button**」または「**link**」の場合、この値はボタンまたはリンク上のテキストとして使用されます。値を指定しない場合は、デフォルトで正符号 **(+)** が使用されます。空ストリング値 "" は使用できず、デフォルトで正符号 **(+)** に設定されます。

expandRepresentation が「**image**」の場合は、この値は使用するイメージ・ファイルのパスおよび名前です。パスは、プロジェクトの Web コンテンツ・ディレクトリーを基準にして検索されます。例えば、Web Content/common/images ディレクトリーに twisty1.gif を指定する場合、次のように expandValue を設定します。

```
<setting name="expandValue" value="/common/images/twisty1.gif "/>
```

expandAltValue

この設定はイメージの代替テキストを提供します。これは、**expandRepresentation** が 「**image**」 に設定され、ブラウザーが指定されたイメージを表示できない場合にのみ使用されます。例えば、Internet Explorer Mobile の「**メニュー**」>「**表示**」>「**画像を表示する**」オプションが選択されていない場合などです。

collapseRepresentation

詳細を隠すコントロールに使用するグラフィカル表現を決定します。「**button**」、「**link**」、または「**image**」を指定します。デフォルトは「**link**」です。

collapseValue

collapseRepresentation が「**button**」または「**link**」の場合、この値はボタンまたはリンク上のテキストとして使用されます。値を指定しない場合は、デフォルトで負符号 **(-)** が使用されます。空ストリング値 "" は使用できず、デフォルトで負符号 **(-)** に設定されます。

collapseRepresentation が「**image**」の場合は、この値は使用するイメージ・ファイルのパスおよび名前です。パスは、プロジェクトの Web コンテンツ・ディレクトリーを基準にして検索されます。例えば、Web Content/common/images ディレクトリーに twisty1.gif を指定する場合、次のように collapseValue を設定します。

```
<setting name="collapseValue" value="/common/images/twisty1.gif "/>
```

collapseAltValue

この設定はイメージの代替テキストを提供します。これは、**collapseRepresentation** が 「**image**」 に設定され、ブラウザーが指定されたイメージを表示できない場合にのみ使用されます。例えば、Internet Explorer Mobile の「**メニュー**」>「**表示**」>「**画像を表示する**」オプションが選択されていない場合などです。

expandHeaderValue

この設定では、詳細コントロールの列のヘッダー・テキストを指定します。"" と指定された値は、テキストのないヘッダーを作成します。

expandStyle、expandClass、expandHeaderStyle、expandHeaderClass、expandAreaStyle、expandAreaClass、expandRowStyle、expandRowClass、collapseStyle、collapseClass、collapseAreaStyle、collapseAreaClass

これらの設定は、当該オプションのコントロールのクラスおよびスタイルのオーバーライドです。指定しない場合は、ウィジェットの標準値が適用されます。標準 HTML スタイルを使用でき、クラスは順番に適用されます。

- 「**入力フィールドのカーソル位置オプションを使用可能にする**」設定は、入力フィールドのレンダリングに使用するウィジェットで使用でき、ユーザーはこの設定を使用して、入力フィールドでデータ入力モードからカーソル位置決めモードに切り替えることができます。データ入力モードでは、ユーザーは入力フィールドにデータを入力できます。カーソル位置決めモードでは、ユーザーはタブまたはその他の方法により、入力フィールド内の任意の文字上にカーソルを置くことができます。この設定は、他のカーソル位置決め機能がない装置を使用する場合に役立ちます。

  258 ページの『ウィジェット設定』で説明されている「**入力フィールドのカーソル位置オプションを使用可能にする**」 ウィジェット設定に加えて、次の設定を使用して、このオプションで使用するコントロールの外観をさらにカスタマイズできます。これらの設定は、HATS Toolkit GUI には表示されません。代わりに、ウィジェットのソースにこれらの設定を追加する必要があります。以下のソースは、例としてフィールド・ウィジェットを使用したときのこれらの設定のサンプルです。

```
<class name="com.ibm.hats.transform.widgets.FieldWidget">
     <setting name="dataModeCEPRepresentation" value="button"/
     <setting name="dataModeCEPValue" value="back"/>
     <setting name="dataModeCEPAltValue" value="data entry"/>
     <setting name="cursorModeCEPRepresentation" value="image"/>
     <setting name="cursorModeCEPValue" value="cursorOn.gif"/>
     <setting name="cursorModeCEPAltValue" value="cursor entry"/>
     <setting name="cursorCEPRepresentationStyle"
              value="text-underline:true"/>
     <setting name="useCursorExactPositioningOption" value="true"/>
</class>
```

dataModeCEPRepresentation

データ・モードの場合に、入力フィールドの横に表示されるコントロールに使用するグラフィカル表現を決定します。「**button**」、「**link**」、または「**image**」を指定します。デフォルトは「**link**」です。

dataModeCEPValue

dataModeCEPRepresentation が「**button**」または「**link**」の場合、この値はボタンまたはリンク上のテキストとして使用されます。値を指定しない場合は、デフォルトでアスタリスク文字 (*) が使用されます。空ストリング値 "" は使用できず、デフォルトでアスタリスク文字 (*) に設定されます。

dataModeCEPRepresentation が「**image**」の場合は、この値は使用するイメージ・ファイルのパスおよび名前です。パスは、プロジェクトの Web コンテンツ・ディレクトリーを基準にして検索されます。例えば、Web Content/common/images ディレクトリーに sbarleftblk.gif を指定する場合、次のように dataModeCEPValue を設定します。

```
<setting name="dataModeCEPValue"
        value="common/images/sbarleftblk.gif"/>
```

dataModeCEPAltValue

この設定はイメージの代替テキストを提供します。これは、dataModeCEPRepresentation が 「**image**」 に設定され、ブラウザーが指定されたイメージを表示できない場合にのみ使用されます。例えば、Internet Explorer Mobile の「**メニュー**」>「**表示**」>「**画像を表示する**」オプションが選択されていない場合などです。

dataModeIconClass、dataModeIconStyle

これらの設定は、データ・モード表示アイコンのスタイルおよびクラスのオーバーライドです。指定しない場合は、ウィジェットの標準値が適用されます。標準 HTML スタイルを使用でき、クラスは順番に適用されます。

cursorModeCEPRepresentation

カーソル・モードの場合に、入力フィールドの横に表示されるコントロールに使用するグラフィカル表現を決定します。「**button**」、「**link**」、または「**image**」を指定します。デフォルトは「**link**」です。

cursorModeCEPValue

cursorModeCEPRepresentation が「**button**」または「**link**」の場合、この値はボタンまたはリンク上のテキストとして使用されます。値を指定しない場合は、デフォルトでアスタリスク文字 (*) が使用されます。空ストリング値 "" は使用できず、デフォルトでアスタリスク文字 (*) に設定されます。

cursorModeCEPRepresentation が「**image**」の場合は、この値は使用するイメージ・ファイルのパスおよび名前です。パスは、プロジェクトの Web コンテンツ・ディレクトリーを基準にして検索されます。例えば、Web Content/common/images ディレクトリーに sbarrightblk.gif を指定する場合、次のように cursorModeCEPValue を設定します。

```
<setting name="cursorModeCEPValue"
        value="common/images/sbarrightblk.gif "/>
```

cursorModeCEPAltValue

この設定はイメージの代替テキストを提供します。これは、**cursorModeCEPRepresentation** が 「**image**」に設定され、ブラウザーが指定されたイメージを表示できない場合にのみ使用されます。例えば、Internet Explorer Mobile の「**メニュー**」>「**表示**」>「**画像を表示する**」オプションが選択されていない場合などです。

cursorCEPRepresentationStyle

このスタイルは、現在選択されているカーソル位置を強調表示する場合に使用されます。デフォルトは「**border-bottom: 1px solid green;**」です。ただし、空ストリング "" は有効です。

cursorModeCEPStyle

これは、カーソル・モードの場合に、入力フィールドで使用されるスタイルです。デフォルトのスタイルは、「**border: 1px solid #999999; height: 1.75em;**」です。必要な場合以外は、このスタイルを変更しないようにしてください。

cursorCEPRepresentationClass、cursorModeIconClass、cursorModeIconStyle、cursorModeCEPClass、cursorModeCEPTextStyle、cursorModeCEPTextClass
これらの設定は、カーソル・モード領域の要素を修飾するためのクラスおよびスタイルのオーバーライドです。指定しない場合は、ウィジェットの標準値が適用されます。標準 HTML スタイルを使用でき、クラスは順番に適用されます。

dataModeIconClass、dataModeIconStyle
これらの設定は、データ・モード領域の要素を修飾するためのクラスおよびスタイルのオーバーライドです。指定しない場合は、ウィジェットの標準値が適用されます。標準 HTML スタイルを使用でき、クラスは順番に適用されます。

- ワイヤレス接続の際に、ルーターまたは HTTP サーバーへの接続が失われた、複数のインスタンスがあります。これが発生すると、Web ページがロックされることがあり、HATS で情報を再送できません。この問題を解決するために、以下の設定を使用できます。これらの設定は、「プロジェクト設定」ソース・ビュー (application.hap ファイル) の RuntimeSettings クラスで使用できます。

  usePageUID
  pageUID の画面配置を有効にするかどうかを指定します。これにより、ランタイムで、ページが送信済みであるかどうかを追跡できます。この値が false の場合は、pageSubmitTimeout 値が読み取られません。デフォルト値は **false** です。

  pageSubmitTimeout
  Web ページのロックを解除してユーザーが使用できるようにするまでの待ち時間をミリ秒単位で指定します。値に -1 を指定するか空にすると、この機能が非アクティブになります。デフォルト値は **-1** です。

  incorrectPageUIDEvent
  ユーザーの Web ページによる HTTP 要求の PageUID が、該当のセッション ID およびアプリケーション・インスタンスのランタイムに保管されている PageUID と一致しない場合の対処方法を指定します。
  REFRESHCOMMAND または ERROREVENT の 2 つのオプションがあります。デフォルトは **REFRESHCOMMAND** です。

  以下のソースは、これらの設定のサンプルを示しています。

```
<class name="com.ibm.hats.common.RuntimeSettings">
   <setting name="autoEraseFields" value="true"/>
   <setting name="enableAutoAdvance" value="false"/>
   <setting name="enableAutoTabOn" value="false"/>
   <setting name="enableBusyPage" value="false"/>
   <setting name="enableCompression" value="false"/>
   <setting name="enableOverwriteMode" value="false"/>
   <setting name="incorrectPageUIDEvent" value="REFRESHCOMMAND"/>
   <setting name="pageSubmitTimeout" value="1000"/>
   <setting name="selectAllOnFocus" value="false"/>
   <setting name="suppressUnchangedData" value="false"/>
   <setting name="usePageUID" value="true"/>
</class>
```

- iPhone または iPod touch デバイスで実行されているブラウザーから別のアプリケーションに切り替えると、ブラウザーからの AJAX ポーリングが停止します。この結果、HATS にアクセスしているブラウザーから、デバイス上の別のアプリケーションに切り替えたとき、**切断を待機する時間 (秒)** 間隔が経過した後、

HATS によりブラウザー・セッションが切断されます。この間隔を長くするのが適切な場合、あるいは iPhone または iPod touch デバイスからアクセスされる HATS アプリケーションの自動切断機能を無効にするのが適切な場合があります。詳しくは、 134 ページの『クライアント・プル (AJAX) メソッドの使用』を参照してください。

- DBCS の考慮事項について詳しくは、 549 ページの『モバイル装置アプリケーションの操作』を参照してください。

HATS が自動的に使用不可に設定するオプションの他に、以下の機能はサポートされていないため、モバイル装置用の HATS アプリケーションにはインプリメントしないでください。

- HATS Web プロジェクトをモバイル装置用 HATS Web プロジェクトにマイグレーションまたは自動変換することはできません。
- リッチ・クライアント・アプリケーション。
- ポータル・アプリケーション (標準または IBM)。
- WebFacing との相互運用性。
- 双方向言語サポート。
- アクセシビリティー機能。
- 320x240 より小さい画面サイズのあらゆる装置。
- キーボード・ホスト・キー・サポート。

  その結果、キーボードの使用は通常の Web キーボードと同様に機能しますが、ホスト・キーは送信されず、代わりに F1 キーでブラウザー・ヘルプを起動します。また、Enter キーはホスト Enter キーにマップされません。

  また、キーボード・サポートがないため、HATS JavaScript はユーザーが HATS Web ページにデータを入力したことを判別できません。このため、新しいホスト画面がホスト・アプリケーションから非同期で受信されると、ユーザーが HATS アプリケーションの GUI ビューに入力したデータを自動最新表示サポートが上書きします。
- フィールド固有のヘルプ、右寄せ、先頭文字の大文字化、またはその他の属性。
- 装置のハードウェアまたはソフトウェアが原因の、その他のあらゆるデータ入力制限。
- モバイル装置による HATS 管理コンソール・ページへのアクセス。
- disconnectOnClose 接続パラメーター。
- 画面組み合わせ。
- Windows Mobile のカレンダー・ウィジェット。
- タブ付きフォルダーのサポート。
- スプレッドシートのサポート。
- Dojo ウィジェット。

iPad デバイスに関する考慮事項および制約事項

iPad デバイスの場合、他のモバイル・デバイスと比較して、画面サイズや Web ブラウザーの違いにより、HATS アプリケーションの表示や動作が異なることがあります。 iPad デバイスには、他の一部のモバイル・デバイスと比べて大きい画面が備

わっているため、モバイル・デバイスに最適化されたオプションで作成されていないアプリケーションも iPad デバイスではより快適に使用できます。 iPad デバイス用の HATS Web プロジェクトを作成する際には、**「モバイル装置用にオプションを最適化」**オプションを選択しないでください。

最新のサポートに関する考慮事項については、「Host Access Transformation Services 9.5 - Known issues and workarounds」(http://www.ibm.com/support/docview.wss?uid=swg27046998) を参照してください。

Android デバイスに関する考慮事項および制約事項

Android デバイスの場合、他のモバイル・デバイスと比較して、画面サイズや Web ブラウザーの違いにより、HATS アプリケーションの表示や動作が異なることがあります。ご使用の Android デバイスの画面サイズが小さい場合は、そのデバイスで使用するために、HATS Web プロジェクトではなく HATS モバイル・プロジェクトを作成することを考慮してください。これにより、小さい画面でのユーザー・エクスペリエンスが向上します。詳しくは、53 ページの『モバイル装置用の HATS アプリケーションの作成』を参照してください。

最新のサポートに関する考慮事項については、「Host Access Transformation Services 9.5 - Known issues and workarounds」(http://www.ibm.com/support/docview.wss?uid=swg27046998) を参照してください。

第 4 章 HATS リッチ・クライアント・アプリケーションの開発とデプロイ

HATS は、Eclipse リッチ・クライアント・プラットフォーム (Eclipse RCP) 環境、Lotus Notes 環境、または Lotus Expeditor Client 環境で実行するリッチ・クライアント GUI アプリケーションを使用して、既存のホスト・アプリケーションへのアクセスを変換する機能を提供します。

Eclipse RCP は、Eclipse プラットフォーム (オープン・ソースのツール・プラットフォームとして最も良く知られる) によって提供されているプラグインのサブセットです。Eclipse RCP の導入により、Eclipse で提供されるコア機能を使用して、ユーザーのデスクトップを対象としたネイティブ・クライアント・アプリケーションを構築することが可能になります。Eclipse RCP の詳細情報は、http://www.eclipse.org/home/categories/rcp.php にあります。

Lotus Notes は強力なマルチ・ファセットのソフトウェアであり、ユーザーにとって重要な情報すべてに簡単にアクセスできます。Lotus Notes を使用して、インターネット・メールの送受信、予約のスケジュール、Web の参照、インターネット・ニュースグループへの参加、およびホーム・ページを活用した日々の重要な情報すべてのトラッキングを行うことができます。詳しくは、Lotus Notes Knowledge Center (http://publib.boulder.ibm.com/infocenter/domhelp/v8r0/index.jsp) を参照してください。

Lotus Expeditor Client は、多数のエンタープライズ・アプリケーションをサーバーが管理するラップトップおよびデスクトップ・システムに拡張するためのリッチ・クライアント・ランタイム環境および統合されたミドルウェア・コンポーネントを提供します。Lotus Expeditor Client のアプリケーションの開発やデプロイの詳細については、Lotus Expeditor Knowledge Center (http://publib.boulder.ibm.com/infocenter/ledoc/v6r2/index.jsp) を参照してください。

注:

特に明記しない限り、本章での説明は、Eclipse RCP V3.7 環境、Lotus Notes V8.5.3 環境、および Lotus Expeditor Client V6.2.3 環境での HATS リッチ・クライアント・アプリケーションの開発と展開に適用されます。前のバージョンの Eclipse RCP および Lotus Expeditor Client に適用される説明については、前のバージョンの適切な Knowledge Center を参照してください。

リッチ・クライアント環境は、従来の Web 環境では使用できない、以下のような機能を提供します。

- 向上した応答時間。
- より豊富なユーザー・インターフェース (UI) ウィジェットのセット。アプリケーションのより固有の外観を提供します。例えば、固有のタブ・フォルダー・コントロール、編集可能コンボ・ボックス、ツールバー、メニュー・バー、およびテーブルなどがあります。
- WebSphere Application Server または WebSphere Portal との依存関係がないこと。

© Copyright IBM Corp. 2003, 2015

- クライアント・サイドの処理 (単一サーバー上に集中するのではなく、分散される)。
- 3270E 印刷ジョブをユーザーのローカル・プリンターに直接印刷する。

リッチ・クライアント・アプリケーションの開発に関する追加情報およびヘルプは、Rational SDP のヘルプ目次にあります。メニュー・バーの「**ヘルプ**」>「**ヘルプ目次**」を選択して、「**リッチ・クライアント**」 を検索します。

HATS リッチ・クライアント・アプリケーションを作成する場合、Eclipse RCP 環境、Lotus Notes 環境、および Lotus Expeditor Client 環境の間の違いはごくわずかです。この章全体でその違いについて説明します。

HATS リッチ・クライアント・アプリケーションと HATS Web アプリケーションの間の主なアーキテクチャーの違いは、Telnet ソケット接続をどこで開始するかという点です。リッチ・クライアント・アプリケーションを使用する場合、従来の HATS Web アプリケーションと同様に、ソケット接続は WebSphere Application Server ではなく、ユーザーのワークステーションから開始します。リッチ・クライアント・アプリケーションは、ユーザーのワークステーション上にクライアントを必要とするため、これらのアプリケーションは主に、従来のファット・クライアント端末エミュレーターの使用に慣れた内部ユーザーを対象としています。

HATS リッチ・クライアント・アプリケーションの開発

HATS リッチ・クライアント・アプリケーションを構築およびテストするためのステップは、基本的に HATS Web アプリケーションの場合と同じです。 HATS Toolkit を HATS パースペクティブで提供されるツールとともに使用して、HATS リッチ・クライアント・プロジェクトを開発し、そこから HATS リッチ・クライアント・アプリケーションが生成されます。開発中に必要な様々なアクティビティーをサポートするために、同じウィザード、エディター、およびビューが提供されています。

Web 環境用の HATS と同様に、アプリケーションの外観および操作方法を開発します。これには、デフォルトのレンダリング設定、画面ナビゲーション (skip screen マクロを使用)、および画面のレンダリングに使用されるフォントと色のカスタマイズが含まれます。

HATS Web アプリケーションで使用可能なフィーチャーの大多数は、HATS リッチ・クライアント・アプリケーションでも使用可能です。これには、以下のものが含まれますが、それに限定されるわけではありません。

- デフォルト・レンダリング
- グローバル規則
- テキスト置換
- カスタム画面変換
- 定義済みの変換
- グローバル変数
- カスタム・コンポーネントおよびウィジェット
- マクロおよびマクロ・ハンドラー

- バックグラウンド接続
- Java ビジネス・ロジック
- アプリケーション・レベルのイベント。例えば、開始、停止、接続、および切断。
- キーボード・サポート
- 非同期更新
- Web プロジェクトの「サーバーで実行」に類似したローカル・テスト環境

HATS リッチ・クライアント・アプリケーションでは使用できないが、HATS Web アプリケーションでは使用可能なフィーチャーは以下のとおりです。

- 統合オブジェクトの作成
- EJB Access Bean の作成
- Web 高速ログオン (WEL)
- HATS 管理コンソール

HATS リッチ・クライアント・プロジェクトは、Eclipse プラグイン・プロジェクトです。これはプロジェクトとしてワークスペース内にあり、HATS アプリケーション開発者によって開発されます。プロジェクトが開発およびテストされると、HATS リッチ・クライアント・アプリケーションとしてエクスポートされ、これが、Eclipse 環境にインストールできる Eclipse プラグインとなります。Eclipse プラグインは、Eclipse 環境のコンテキスト内で特定のタイプのサービスを提供するコンポーネントです。HATS で使用する場合、プラグインはホスト・アプリケーションを変換するために必要なサービスを提供します。

それぞれの HATS リッチ・クライアント・アプリケーション・プラグインは、複数のランタイム・プラグインに依存します。依存関係はリッチ・クライアント・プロジェクトのマニフェスト・ファイル、MANIFEST.MF で参照されます。HATS リッチ・クライアント・プラグインが正しく実行されるためには、これらの必須プラグインがローカル・テスト環境および展開済みの実稼働環境に組み込まれていることを確認する必要があります。詳しくは、 69 ページの『HATS RCP ランタイム拡張プロジェクト』、 74 ページの『HATS リッチ・クライアント・アプリケーションのテスト』、および 80 ページの『HATS リッチ・クライアント・アプリケーションのデプロイ』を参照してください。

ターゲット・プラットフォームの特性

プロジェクト・コンテンツ

HATS リッチ・クライアント・プロジェクトを作成する場合、ターゲット・プラットフォームを選択するように求めるプロンプトが出されます。この選択は、アプリケーションの展開先を示します。Eclipse RCP への展開を対象としたプロジェクトと Lotus Notes または Lotus Expeditor Client への展開を対象としたプロジェクトの間の違いはごくわずかです。以下の特性は、すべての HATS リッチ・クライアント・プロジェクトに共通です。

- プロジェクト構造 (テンプレート、変換、イメージ、Java ソース、マクロ・ハンドラー、マクロ、および画面キャプチャー・フォルダーを含む)。

- HATS 固有のプロジェクト成果物 (application.hap ファイル、接続ファイル、およびイベント・ファイルを含む)。
- ComponentWidget.xml ファイル。これは、コンポーネントおよびウィジェット・レジストリー・ファイルです。
- plugin.xml ファイル。この中に 1 つのビュー、つまり変換ビューが登録されます。
- ランタイム・プラグインでのプラグイン依存関係。
- Perspective クラス。HATS RCP ランタイム拡張 プロジェクトに組み込まれており、ページ上でビューをレイアウトする役割を担います。 69 ページの『HATS RCP ランタイム拡張プロジェクト』を参照してください。

Eclipse RCP への展開を対象としたプロジェクトの追加事項:

- Application クラス。Eclipse RCP ではアプリケーションを起動する必要があるため、HATS RCP ランタイム拡張プロジェクトに組み込まれています。 69 ページの『HATS RCP ランタイム拡張プロジェクト』を参照してください。

Lotus Notes および Lotus Expeditor Client への展開を対象としたプロジェクトの追加事項:

- HATS RCP ランタイム拡張プロジェクトの plugin.xml にある、アプリケーション・スイッチャー項目の登録。これにより、ユーザーは Lotus Notes 環境および Lotus Expeditor Client 環境のアプリケーション・スイッチャーから HATS アプリケーションを起動することができます。
- HATS RCP ランタイム拡張 プロジェクトの plugin.xml にある、Lotus Notes プラットフォームのキー・バインディング。これにより、ユーザーはキー・マッピングを Lotus Notes 環境で使用するように定義できます。

コンパイラー準拠レベルの設定

Rational SDP ワークスペースと意図しているランタイム環境の両方の Java ランタイム環境 (JRE) が、同じである必要があります。例えば、Java ランタイム環境 (JRE) 1.6 を実行する Eclipse 環境で稼働するように HATS リッチ・クライアント・プラグインを開発している場合は、ワークスペースのコンパイラー準拠レベルを 1.6 に設定する必要があります。これを行うには、Rational SDP メニュー・バーから、「**ウィンドウ**」>「**設定**」>「**Java**」>「**コンパイラー**」を選択し、「**コンパイラー準拠レベル**」を 1.6 に設定します。

他の準拠レベルを必要とする他の Java またはプラグイン・プロジェクトがワークスペース内にある場合、個々の Java またはプラグイン・プロジェクトのコンパイラー準拠レベルを設定することができます。これを行うには、プロジェクトを右クリックし、「**プロパティー**」を選択します。「プロパティー」パネルで、左側の「**Java コンパイラー**」を選択します。次に、「**プロジェクト特定設定を使用可能にする**」ボックスを選択し、「**コンパイラー準拠レベル**」を適切なレベルに設定します。

注: HATS RCP ランタイム拡張プロジェクト com.ibm.hats.rcp.runtime.extension は、HATS リッチ・クライアント・プロジェクトと同じコンパイラー準拠レベルでなければなりません。 69 ページの『HATS RCP ランタイム拡張プロジェクト』を参照してください。

HATS RCP ランタイム拡張プロジェクト

HATS リッチ・クライアント・プロジェクトを作成すると、HATS RCP ランタイム拡張プラグイン (以後、ランタイム拡張プラグイン) のプロジェクトは、ワークスペース内にまだ存在していない場合には自動的に作成されます。ランタイム拡張プラグインには、com.ibm.hats.rcp.runtime.extension の名前とプラグイン ID があり、初期バージョンは 1.0.0 です。

ランタイム拡張プラグインは、HATS RCP ランタイム (com.ibm.hats.rcp.runtime) プラグインの拡張です。HATS RCP ランタイム・プラグインには HATS ランタイム・クラスが含まれており、HATS リッチ・クライアント・アプリケーションが機能するには、その環境内になければなりません。ランタイム拡張プラグインにより、ランタイム環境に関連した設定を行うことができます。

このプラグイン・プロジェクトには、以下の構成可能な成果物が含まれています。

- runtime.properties ファイル - ランタイム環境および「実行」モードで稼働するローカル・テスト環境で、HATS によって使用されるログ、トレース、および他の設定を指定するために使用されます。これらの設定の詳細については、93 ページの『HATS リッチ・クライアント・アプリケーションの管理』を参照してください。
- runtime-debug.properties ファイル - 「デバッグ」モードで稼働するローカル・テスト環境で、HATS によって使用されるランタイム設定を指定するために使用されます。これらの設定の詳細については、93 ページの『HATS リッチ・クライアント・アプリケーションの管理』を参照してください。
- plugin.xml ファイル - 「アプリケーション」ビューを環境に登録し、初期キーボード・マッピングを定義します。
- product.xml ファイル - HATS バージョンを識別します (HATS EAR プロジェクト内のファイルと同じ)。

このプロジェクトは Web 環境の HATS EAR プロジェクトに類似していますが、ランタイム環境ごとにこれらのプラグインのうちの 1 つしかありません。つまり、ユーザーの環境では、ランタイム拡張プラグインが 1 つだけ稼働していることになります。すべての HATS リッチ・クライアント・プロジェクト plugin.xml ファイルは、このプラグインで依存関係を宣言します。ランタイム拡張プラグインは、個々の HATS アプリケーション・プラグインを参照しません。

ランタイム環境で実行できるランタイム拡張プラグインのコピーは 1 つだけなので、ユーザーの環境にはプラグインのコピーが 1 つだけ存在するようにする必要があります。複数のランタイム拡張プラグインが環境内にある場合は、バージョン番号が最も大きいプラグインがその環境によって使用されます。これによって、最も新しいコードが常に使用されます。

ランタイム拡張プロジェクトは、「**HATS Projects**」ビューには表示されませんが、「**ナビゲーター**」ビューには表示されます。これは、エクスポート時に生成されるフィーチャー・プロジェクトの一部として組み込まれます。詳しくは、80 ページの『HATS リッチ・クライアント・アプリケーションのデプロイ』を参照してください。

注: このプロジェクトは、最新の HATS リッチ・クライアント・プロジェクトが削除されるときには削除されません。新規のランタイム拡張プロジェクトを作成する場合は、手動で削除する必要があります。

HATS リッチ・クライアント・プロジェクトの処理

HATS Web プロジェクトと同じように、新規 HATS リッチ・クライアント・プロジェクトを作成すると、HATS ファイルを編成するために利用できる一組のフォルダーが作成されます。以下にデフォルト・プロジェクトの例を示します。 最上位のフォルダーの名前は、作成するプロジェクトに付けた名前と同じです。このフォルダーの中には、HATS プロジェクトで定義したオブジェクトを含む他の上位フォルダーが入っています。一部のフォルダーは、特定のオブジェクトが作成されるまで表示されません。

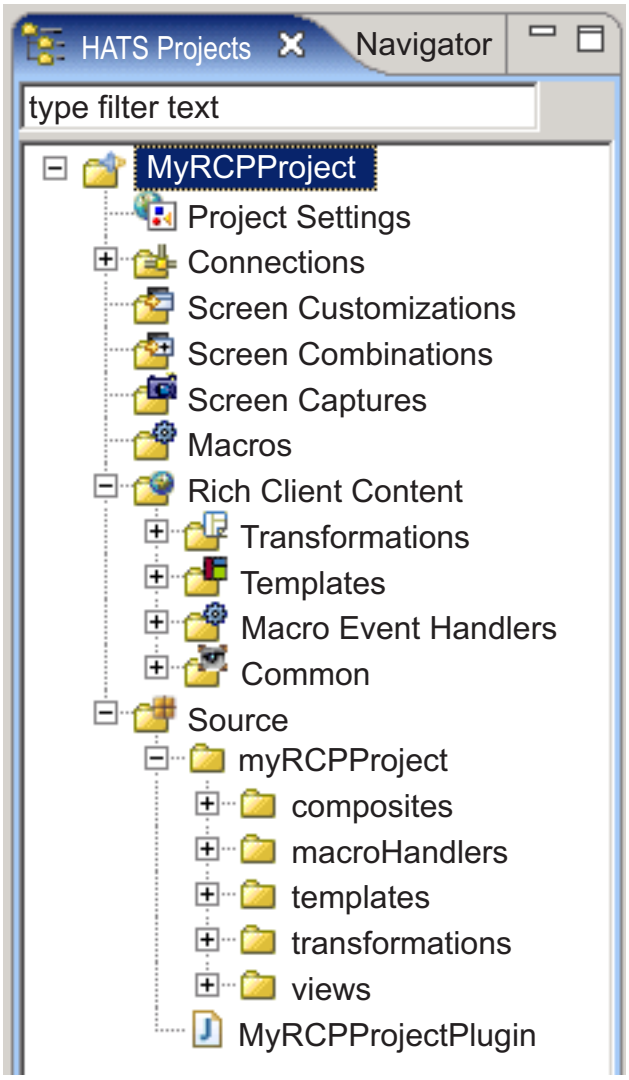


図 6. 「HATS リッチ・クライアント・プロジェクト」ビュー

プロジェクトのセットアップ方法に応じて、これらのフォルダーの一部またはすべてが「**HATS Projects**」ビューに表示されます。また、「**HATS Projects**」ビューに

表示するフォルダーや、非表示にするファイル拡張子を指定することもできます。詳しくは、 149 ページの『HATS プリファレンスの使用』を参照してください。

注:

1. リッチ・クライアント・プロジェクトは「**リッチ・クライアント・コンテンツ**」フォルダーを表示します。このフォルダーで、Web プロジェクトは「**Web コンテンツ**」フォルダーを表示します。
2. 「**リッチ・クライアント・コンテンツ**」フォルダーは、リッチ・クライアント・プロジェクトの root レベルに表示されます。ただし、プロジェクトには実際の「**リッチ・クライアント・コンテンツ**」フォルダーはありません。Web プロジェクト用に表示される「**Web コンテンツ**」フォルダーとの整合性を作成するためにのみ、1 つのフォルダーが動的に表示されます。
3. 動的に表示された「**リッチ・クライアント・コンテンツ**」フォルダーの下に表示されるフォルダーのいくつかは「**ソース**」フォルダーの下にも表示されます。
4. 動的に表示されたリッチ・クライアント・フォルダー用のラベル・デコレーターが正しく装飾しないことがあります。これは、それらのフォルダーがプロジェクト内の物理フォルダーにマップしていないためです。

プロジェクトのファイルの編成をわかりやすくするために、これらの上位サブフォルダーの内部にもフォルダーを作成することができます。例えば、プロジェクト用の画面キャプチャーを作成するときに、「**画面キャプチャー**」フォルダーの下に、取り込んだ画面を編成してグループ化するためのフォルダーを作成することができます。フォルダーを作成するには、ツリー内の上位フォルダーの 1 つを右マウス・ボタンでクリックし、「**新規 HATS**」>「**フォルダー**」を選択します。ファイルを別のフォルダーに移すには、該当ファイルを右マウス・ボタンでクリックして「**移動**」を選択するか、ドラッグ・アンド・ドロップ方式を使用します。

HATS パースペクティブの「**ナビゲーター**」**ビュー**に進むと、HATS プロジェクトをチーム環境で共用できます。プロジェクトを右クリックして、「**チーム**」>「**プロジェクトの共用**」を選択します。リストからリポジトリー・タイプを選択し、「**次へ**」をクリックします。Rational SDP は、いくつかのリポジトリーをサポートしています。詳しくは、Rational SDP の資料を参照して「**リポジトリー**」を検索してください。

注: HATS プロジェクトでバージョン管理システムを使用する場合は、`resourceUpdate.sts` ファイルを無視するようにバージョン管理システムを設定します。このファイルは、ツールキット内でプロジェクトをテストするときに自動的に生成されます。このファイルはバージョン管理では不要であり、バージョン管理下で HATS プロジェクトを設置する前に安全に無視したり削除したりできます。

このファイルをバージョン管理から除外するには、「**ウィンドウ**」->「**設定**」->「**チーム**」->「**無視するリソース (Ignored Resources)**」を開いて「**パターンの追加 (Add Pattern)**」クリックし、新規パターンを追加します。`resourceUpdate.sts` と入力して、「**OK**」をクリックします。無視するパターンのリストで新規パターンが選択されていることを確認し、「**OK**」をクリックして設定を保存します。

Web プロジェクトとリッチ・クライアント・プロジェクトの間でのリソースのコピー

「HATS Projects」ビューのポップアップ・メニューを使用して、リッチ・クライアント・リソースのコピー、貼り付け、削除、移動、および名前変更を行うことができます。以下のリソースを、HATS Web プロジェクトと HATS リッチ・クライアント・プロジェクトの間でコピーまたは移動できます。

- 接続プロファイル
- マクロ
- 画面キャプチャー
- 画面カスタマイズ
- イメージ
- Java ソース・ファイル
- BMS ファイル
- ホスト・シミュレーション・ファイル

注: 変換、変換フラグメント、またはテンプレートを HATS Web プロジェクトと HATS リッチ・クライアント・プロジェクトの間でコピーすることは許可されていません。

異なるタイプのプロジェクトからコピーされたリソースによっては、変更を必要とするものがあります。例えば、HATS Web プロジェクトから HATS リッチ・クライアント・プロジェクトにコピーされたマクロは、有効な SWT Java マクロ・ハンドラーを参照するように更新される必要があります (HATS Web プロジェクトの場合、マクロは JSP マクロ・ハンドラーを参照します)。また、適用アクションで指定された JSP 変換の参照や転送アクションで指定された URL により、画面カスタマイズを更新する必要もあります。問題を検出し、修正するために、警告メッセージが「**問題**」ビューに表示されます。以下のシナリオには警告が出されます。

- 正しくない成果物を参照する画面カスタマイズ・アクション (通常、プロジェクトに存在しない変換を参照する適用アクション)。
- 正しくないマクロ・ハンドラーを参照するマクロ。
- リッチ・クライアント画面カスタマイズの転送またはブロック・アクション。

HATS Web アプリケーションを HATS RCP アプリケーションにマイグレーションするための組み込み機能はありません。画面のカスタマイズ、マクロ、接続、画面キャプチャー、イメージ、ビジネス・ロジック・ソース・ファイル、およびカスタム・コンポーネント・ソース・ファイルなどの成果物は、最小限の変更を加えるだけで 1 つのプロジェクト・タイプから別のプロジェクト・タイプに手動でマイグレーション可能です。変換、カスタム・ウィジェット、テンプレート、およびカスタム・マクロ・ハンドラーなどの他の成果物は、異なるタイプのプロジェクトで使用するためにより多くの変更を必要とします。

HATS リッチ・クライアント・プロジェクトのエクスポートとインポート

HATS リッチ・クライアント・プロジェクトを Rational SDP がインストールされている別のシステムに移動するか、またはその他の理由でプロジェクトを保存および復元するには、Rational SDP のエクスポートおよびインポート機能を使用できます。

エクスポート

Rational SDP のアーカイブ・ファイルを使用して、プロジェクトとその従属プロジェクトを .zip ファイルとしてエクスポートします。この機能には、複数の特長があります。この機能では、複数のプロジェクトと必要なランタイム拡張プロジェクト (com.ibm.hats.rcp.runtime.extension) をすべて同時にエクスポートできます。この場合、アーカイブ・ファイルをインポートする前に、宛先の HATS Toolkit でプロジェクトを作成する必要はありません。HATS Toolkit からプロジェクトをアーカイブ・ファイルとしてエクスポートするには、次の手順を実行します。

1. メニュー・バーから、「**ファイル**」>「**エクスポート**」をクリックして、「エクスポート」ウィザードを開きます。
2. 「**一般**」>「**アーカイブ・ファイル**」を選択して、「**次へ**」をクリックします。
3. エクスポートするプロジェクトを選択します。

   注: HATS リッチ・クライアント・プロジェクトでは、実行するためにランタイム拡張プロジェクト (com.ibm.hats.rcp.runtime.extension) が必要です。そのため、ランタイム拡張プロジェクトをアーカイブ・ファイルの一部としてエクスポートするか、または新規の HATS リッチ・クライアント・プロジェクトを作成することによって宛先システム上で新規のランタイム拡張プロジェクトを作成する必要があります。
4. プロジェクトを保存するファイルの名前と場所を指定します。
5. 「**オプション**」セクションで、必要なファイル・フォーマットと圧縮オプションを選択します。プロジェクト全体をエクスポートするには、「**ファイルのディレクトリー構造を作成 (Create directory structure for files)**」を選択します。
6. 「**完了**」をクリックします。

インポート

アーカイブ・ファイルを (または以前のリリースの Rational SDP からプロジェクト交換ファイルを) HATS Toolkit にインポートするには、以下の手順を実行します。

1. メニュー・バーから、「**ファイル**」>「**インポート**」をクリックして、「インポート」ウィザードを開きます。
2. 「**一般**」>「**既存プロジェクトをワークスペースに (Existing Projects into Workspace)**」を選択して、「**次へ**」をクリックします。
3. 「**アーカイブ・ファイルの選択 (Select archive file)**」オプションを選択し、「**参照**」をクリックしてアーカイブ・ファイルを参照します。
4. 「**プロジェクト (Projects)**」セクションで、インポートする 1 つ以上のプロジェクトを選択します。

   注: HATS リッチ・クライアント・プロジェクトでは、実行するためにランタイム拡張プロジェクト (com.ibm.hats.rcp.runtime.extension) が必要です。そのた

め、アーカイブ・ファイルからランタイム拡張プロジェクトをインポートするか、新規の HATS リッチ・クライアント・プロジェクトを作成することによって新規のランタイム拡張プロジェクトを作成する必要があります。

5. 「**完了 (Finish)**」をクリックする。

HATS リッチ・クライアント・アプリケーションのテスト

HATS リッチ・クライアント・アプリケーション・プラグインをローカル・テスト環境でテストするには、そのプラグインと必要なランタイム・プラグインをターゲット・プラットフォームの 1 つのインスタンスで起動します。

注: ローカル・テスト環境でテストすることに加えて、プロジェクトをアプリケーションとしてエクスポートし、それをランタイム環境で展開して、展開プロセスをテストする必要があります。詳しくは、80 ページの『HATS リッチ・クライアント・アプリケーションのデプロイ』を参照してください。

ターゲット・プラットフォームの構成

ターゲット・プラットフォームは Eclipse 製品 (例えば、Eclipse RCP、Lotus Notes、または Lotus Expeditor Client) を参照し、それに対して開発中のプラグインがコンパイルおよびテストされます。

デフォルトでは、ターゲット・プラットフォームは Rational SDP そのものであるため、Eclipse RCP 用の構成は必要ありません。

Lotus Notes (または Lotus Expeditor) 用にターゲット・プラットフォームを構成するには、次の手順を実行します。

1. Lotus Notes (または Lotus Expeditor) を Rational SDP と同じ開発ワークステーションにインストールします。
2. Rational SDP メニュー・バーから、「**ウィンドウ**」>「**設定**」>「**プラグイン開発 (Plug-in Development)**」>「**ターゲット・プラットフォーム**」を選択して、「**追加**」をクリックします。
3. 「**なし: 空のターゲット定義から開始する (Nothing: Start with an empty target definition)**」を選択して、「**次へ**」をクリックします。
4. 「ターゲット・コンテンツ (Target Content)」ページで、ターゲット名 (例えば、`Lotus Notes` または `Lotus Expeditor`) を入力して、「**追加**」をクリックします。
5. 「コンテンツの追加 (Add Content)」ページで、「**ディレクトリー**」を選択して、「**次へ**」をクリックします。
6. 作成するターゲット定義は、Lotus Notes (または Lotus Expeditor) のプラグイン、共有プラグイン、および HATS ランタイム・プラグインで構成されます。「ディレクトリーの追加 (Add Directory)」ページで、以下の手順に従います。
   a. 「**参照**」をクリックして、Lotus Notes (または Lotus Expeditor) の基本プラグインを参照します。 Lotus Notes の場合は、例えば、`C:¥Program Files¥IBM¥Lotus¥Notes¥framework¥rcp¥eclipse` を参照します。 Lotus Expeditor の場合は、例えば、`C:¥Program Files¥IBM¥Lotus¥Expeditor¥rcp¥eclipse` を参照します。「**完了**」をクリックします。

b. 「ターゲット・コンテンツ (Target Content)」ページで、「**追加**」をクリックします。

c. 「コンテンツの追加 (Add Content)」ページで、「**ディレクトリー**」を選択して、「**次へ**」をクリックします。

d. 「**参照**」をクリックして、Lotus Notes (または Lotus Expeditor) の共有プラグインを参照します。 Lotus Notes の場合は、例えば、`C:¥Program Files¥IBM¥Lotus¥Notes¥framework¥shared¥eclipse` を参照します。 Lotus Expeditor の場合は、例えば、`C:¥Program Files¥IBM¥Lotus¥Expeditor¥shared¥eclipse` を参照します。「**完了**」をクリックします。

e. ここで HATS ランタイム・プラグインを追加することはできません。ユーザーがターゲット・プラットフォームを変更すると、HATS Toolkit によってランタイム・プラグインがコピーされます。プラグインはターゲット・プラットフォームにあらかじめインストールされているわけではありません。これは、この後のステップで発生します。

7. 「ターゲット・コンテンツ (Target Content)」ページで、「**完了**」をクリックします。

8. 「設定」-「ターゲット・プラットフォーム」ページで、新しく作成したターゲット定義のボックスを選択し、これが「**アクティブ**」として指定されていることを確認して、「**OK**」をクリックします。

9. これで、HATS ランタイム・プラグインがターゲット・プラットフォームにインストールされました。ユーザーはターゲット・プラットフォームを再ロードするように求められます。ターゲット・プラットフォームの再ロード方法については、『ランタイム・プラグインのインストール』を参照してください。「**OK**」をクリックします。

ランタイム・プラグインのインストール

HATS リッチ・クライアント・アプリケーション・プラグインがターゲット・プラットフォームのローカル・テスト環境で実行されるには、必要なランタイム・プラグインをターゲット・プラットフォームにインストールする必要があります。

Eclipse RCP ターゲット・プラットフォームは Rational SDP そのものです。 HATS Toolkit は Rational SDP にインストールされるため、Eclipse RCP ターゲット・プラットフォーム用の HATS ランタイム・プラグインは既にインストールされています。このセクションの残りの部分では、Lotus Notes (または Lotus Expeditor) のターゲット・プラットフォームを使用する場合のランタイム・プラグインのインストールについて説明します。

新規プロジェクトを作成して Lotus Notes (または Lotus Expeditor) のターゲット・プラットフォームを選択する場合、または後でターゲット・プラットフォームを変更する場合に、HATS Toolkit は、必要なランタイム・プラグインがインストールされていて、選択したターゲット・プラットフォームで最新レベルになっているかどうかを検出します。そうなっていない場合、ツールキットにより、最新のランタイム・プラグインをインストールするように求めるプロンプトが出されます。このプロンプトは、HATS プリファレンスを使用して使用可能または使用不可にすること

ができます。詳細については、 149 ページの『HATS プリファレンスの使用』を参照してください。以下は、インストールされている必須のランタイム・プラグインです。

- HATS Core (com.ibm.hats.core)
- HATS RCP Runtime (com.ibm.hats.rcp.runtime)
- HATS RCP UI (com.ibm.hats.rcp.ui)
- HATS RCP Documentation (com.ibm.hats.rcp.doc)
- ホスト・シミュレーション (com.ibm.hostsim)
- Host On-Demand Bean (com.ibm.eNetwork.beans.HOD)
- Host Access SSLite (com.ibm.etools.hasslite)
- Terminal Beans (com.ibm.etools.terminal.beans)
- SWT Bidirectional 拡張 (com.ibm.editors.swtbidiextension)
- ホスト画面レンダリング (com.ibm.hsrendering)
- WebFacing Common (com.ibm.etools.iseries.wfcommon)
- EMF サービス・データ・オブジェクト (SDO) (org.eclipse.emf.ecore.sdo)
- CommonJ SDO (org.eclipse.emf.commonj.sdo)

Lotus Notes (または Lotus Expeditor) 用に新規ターゲット・プラットフォームを構成して、最新の HATS ランタイム・プラグインをターゲット・プラットフォームにインストールしたら、 74 ページの『ターゲット・プラットフォームの構成』を参照して、新規プラグインを再ロードする必要があります。これを実行するには、以下のステップを実行します。

1. Rational SDP メニュー・バーから、「**ウィンドウ**」>「**設定**」>「**プラグイン開発 (Plug-in Development)**」>「**ターゲット・プラットフォーム**」を選択します。
2. 「設定 - ターゲット・プラットフォーム (Preferences - Target Platform)」ページで、構成した Lotus Notes (または Lotus Expeditor) のターゲット定義を選択します。「**編集**」をクリックします。
3. 「ターゲット・コンテンツ (Target Content)」ページで、「**追加**」をクリックします。
4. 「コンテンツの追加 (Add Content)」ページで、「**ディレクトリー**」を選択して、「**次へ**」をクリックします。
5. 「ディレクトリーの追加 (Add Directory)」ページで、該当するパスの HATS の場所を参照します。Lotus Notes の場合は、例えば、`C:¥Program Files¥IBM¥Lotus¥Notes¥framework¥rcp¥hats¥eclipse` を参照します。 Lotus Expeditor の場合は、例えば、`C:¥Program Files¥IBM¥Lotus¥Expeditor¥rcp¥hats¥eclipse` を参照します。
6. 「**完了**」を 2 回クリックしてウィザードを閉じます。
7. 「設定」-「ターゲット・プラットフォーム」ページで、「**再ロード**」をクリックした後、「**OK**」をクリックします。

デフォルト JRE の設定

ターゲット・プラットフォームを構成し、ランタイム・プラグインをインストールすることに加えて、Rational SDP でワークスペースのデフォルト JRE を設定する必要が生じる可能性があります。

Eclipse RCP

Eclipse RCP ターゲット・プラットフォームでテストする場合は、デフォルト JRE を **jdk** に設定します。これを実行するには、以下のステップを実行します。

1. Rational SDP メニュー・バーから、「**ウィンドウ**」>「**設定**」>「**Java**」>「**インストール済みの JRE**」を選択します。
2. インストールされた JRE のリストで、**jdk** の隣のボックスを選択します。
3. 「**OK**」をクリックします。

JRE を一致させてください。例えば、Java 1.6 を稼働する Eclipse 環境でテストする場合は、ワークスペースの JRE をランタイム環境の JRE に一致するように更新する必要があります。

Lotus Notes

Lotus Notes ターゲット・プラットフォームでテストを行うには、以下のステップを実行します。

1. Rational SDP のメニュー・バーから「**ウィンドウ**」>「**設定**」>「**Java**」>「**インストール済みの JRE (Installed JREs)**」を選択して「**追加**」をクリックします。
2. 「JRE タイプ (JRE Type)」ページで、「**標準 VM (Standard VM)**」を選択して、「**次へ**」をクリックします。
3. 「JRE 定義 (JRE Definition)」ページで「**ディレクトリー**」をクリックして、Lotus Notes の JRE の場所 (例えば、`C:¥Program Files¥IBM¥Lotus¥Notes¥jvm`) を参照します。「**完了**」をクリックします。
4. 「設定」- 「インストール済みの JRE (Installed JREs)」ページで、新しく追加した JRE を選択して、「**OK**」をクリックします。

Lotus Expeditor Client

Lotus Expeditor ターゲット・プラットフォームでテストを行うには、以下のステップを実行します。

1. Rational SDP のメニュー・バーから「**ウィンドウ**」>「**設定**」>「**Java**」>「**インストール済みの JRE (Installed JREs)**」を選択して「**追加**」をクリックします。
2. 「JRE タイプ (JRE Type)」ページで、「**標準 VM (Standard VM)**」を選択して、「**次へ**」をクリックします。
3. 「JRE 定義 (JRE Definition)」ページで「**ディレクトリー**」をクリックして、Lotus Expeditor Client の JRE の場所 (例えば、`C:¥Program Files¥IBM¥Lotus¥Expeditor¥rcp¥eclispe¥plugins¥com.ibm.rcp.j2se.win32.x86_1.6.0.20101125a-201012221418¥jre`) を参照します。「**完了**」をクリックします。
4. 「設定」- 「インストール済みの JRE (Installed JREs)」ページで、新しく追加した JRE のボックスを選択して、「**OK**」をクリックします。

プロジェクトの起動

プロジェクトを起動するには、起動構成が必要です。ご使用のプロジェクトの 「**実行**」または「**デバッグ**」をクリックすると、HATS により、デフォルトの起動構成が作成されます (以下のセクションの説明を参照してください)。 起動構成は、「実行」または「デバッグ」ダイアログが別のビュー、例えば「パッケージ・エクスプ

ローラー」ビューから開かれているときは、作成されません。デフォルトの起動構成は、「新規プロジェクト」ウィザードまたは「プロジェクト設定」エディターで設定される、プロジェクトのターゲット・プラットフォームに基づいています。デフォルトの起動構成をランタイム・ワークベンチをもはや起動できないポイントに変更すると、起動構成を削除することができ、「**実行**」または「**デバッグ**」を再びクリックすると、HATS により、起動構成が再作成されます。プロジェクトを起動するには、以下のセクションのターゲット・プラットフォームについての説明に従ってください。

Eclipse RCP

プロジェクトを Eclipse RCP 用のローカル・テスト環境で起動するには、以下のステップを実行します。

1. 「HATS Projects」ビューから、プロジェクトを右クリックし、「**実行**」または「**デバッグ**」を選択します。
2. 「**デバッグ**」を選択した場合は、「端末の表示」ダイアログで「**はい**」、「**いいえ**」、「**キャンセル**」で応答して、アプリケーションのテスト中にホスト表示端末を表示するかどうかを指定します。
3. 「実行」(または「デバッグ」) 構成ダイアログで、HATS は hostaccess という名前のデフォルトの起動構成を自動的に作成して構成します。
4. 「**メイン**」タブの「**実行するプログラム**」セクションで、「**プロダクトの実行**」フィールドに **com.ibm.hats.rcp.runtime.extension.product** が表示されていることを確認します。
5. 「**実行**」 (または「**デバッグ**」) をクリックして、デフォルト構成を起動します。
6. Eclipse ワークベンチの新規インスタンスが起動されます。
7. 「ホスト・アクセス」パースペクティブが自動的に開きます。
8. ご使用の HATS アプリケーションが「**アプリケーション**」ビューにリストされます。アプリケーションを開始するには、アプリケーションをダブルクリックするか、または右クリックして「**オープン**」を選択します。
9. 選択されたアプリケーションの新規インスタンスがオープンされ、その変換ビューが表示されます。

Lotus Notes

Lotus Notes を Rational SDP と同じ開発ワークステーションにインストールする必要があります。同時に実行できる Lotus Notes インスタンスは 1 つのみであるため、プロジェクトをテストしようとするときに Lotus Notes が既に実行されていてはなりません。 74 ページの『ターゲット・プラットフォームの構成』を参照してください。

プロジェクトを Lotus Notes 用のローカル・テスト環境で起動するには、以下のステップを実行します。

1. 「HATS Projects」ビューから、プロジェクトを右クリックし、「**実行**」または「**デバッグ**」のいずれかを選択します。
2. 「**デバッグ**」を選択した場合は、「端末の表示」ダイアログで「**はい**」、「**いいえ**」、または「**キャンセル**」で応答して、アプリケーションのテスト中にホスト表示端末を表示するかどうかを指定します。

3. 「実行」(または「デバッグ」) 構成ダイアログで、HATS は hostaccess という名前のデフォルトの起動構成を自動的に作成して構成します。
4. 「**メイン**」タブの「**実行するプログラム**」セクションで、「**プロダクトの実行**」フィールドに **com.ibm.notes.branding.notes** が表示されていることを確認します。
5. 「**メイン**」タブの「**Java ランタイム環境 (Java Runtime Environment)**」セクションで「**ランタイム JRE (Runtime JRE)**」フィールドに **Notes 8.5.1 VM** がリストされていることを確認します。詳しくは、77 ページの『Lotus Notes』を参照してください。
6. 「**実行**」 (または「**デバッグ**」) をクリックして、構成を起動します。
7. Lotus Notes の新規インスタンスを起動します。
8. メイン・ツールバーで「**オープン**」をクリックし、「**ホスト・アクセス**」を選択します。
9. ご使用の HATS アプリケーションが「**アプリケーション**」ビューにリストされます。アプリケーションを開始するには、アプリケーションをダブルクリックするか、または右クリックして「**オープン**」を選択します。
10. 選択されたアプリケーションの新規インスタンスがオープンされ、その変換ビューが表示されます。

Lotus Expeditor Client

Lotus Expeditor を対象とした HATS プラグインをローカルにテストするには、Lotus Expeditor Client for Desktop を Rational SDP と同じ開発ワークステーションにインストールする必要があります。 74 ページの『ターゲット・プラットフォームの構成』を参照してください。

プロジェクトを Lotus Expeditor Client 用のローカル・テスト環境で起動するには、以下のステップを実行します。

1. 「HATS Projects」ビューから、プロジェクトを右クリックし、「**実行**」または「**デバッグ**」を選択します。
2. 「**デバッグ**」を選択した場合は、「端末の表示」ダイアログで「**はい**」、「**いいえ**」、「**キャンセル**」で応答して、アプリケーションのテスト中にホスト表示端末を表示するかどうかを指定します。
3. 「実行」(または「デバッグ」) 構成ダイアログで、HATS は hostaccess という名前のデフォルトの起動構成を自動的に作成して構成します。
4. 「**メイン**」タブの「**実行するプログラム**」セクションで、「**プロダクトの実行**」フィールドに **com.ibm.rcp.platform.personality.branding.DefaultProduct** が表示されていることを確認します。
5. 「**メイン**」タブの「**Java ランタイム環境**」セクションで、「**ランタイム JRE**」フィールドに「**Java SE 6.0 Win32 x86**」がリストされていることを確認します。詳しくは、77 ページの『Lotus Expeditor Client』を参照してください。
6. 「**実行**」 (または「**デバッグ**」) をクリックして、構成を起動します。
7. Lotus Expeditor Client の新規インスタンスを起動します。クライアントが初めて開始するときには、新規パスワードを入力するよう求めるプロンプトが出されます。

8. メイン・ツールバーで「**オープン**」をクリックし、「**ホスト・アクセス**」を選択します。
9. ご使用の HATS アプリケーションが「**アプリケーション**」ビューにリストされます。アプリケーションを開始するには、アプリケーションをダブルクリックするか、または右クリックして「**オープン**」を選択します。
10. 選択されたアプリケーションの新規インスタンスがオープンされ、その変換ビューが表示されます。

HATS リッチ・クライアント・アプリケーションのデプロイ

HATS パッケージをどこで取得するかには関係なく (HATS CD、Web、または別の製品とのパッケージ)、同じバージョンの HATS Toolkit をインストールします。これは、HATS の評価に使用できる使用制限付きバージョンです。取得済みライセンス証書に合わせて実稼働環境のランタイムを完全に使用可能にするには、HATS Toolkit に組み込まれている「ライセンス設定」ウィザードを使用して、ライセンス設定を指定する必要があります。詳しくは、「*HATS スタートアップ・ガイド*」の『HATS ランタイムおよびライセンス設定を有効にする 』を参照してください。

このセクションでは、HATS リッチ・クライアント・アプリケーションを展開するための 2 つの異なる手法を説明します。最初の手法は、HATS リッチ・クライアント・アプリケーションを実行するために必要なすべてを備えた配布用の完全な Eclipse クライアント環境をパッケージ化するために使用できます。2 つ目の手法は、すでに Eclipse 環境がインストール済みのクライアント用に HATS リッチ・クライアント・アプリケーションをパッケージ化するために使用できます。

配布用の Eclipse クライアント環境のパッケージ化

Rational SDP のベースとなる Eclipse SDK では、HATS アプリケーション・プラグインと他のすべての必要なプラグインを組み込んだ完全な Eclipse クライアントを生成することができます。このクライアントは、個々のクライアント・システムに配布し、スタンドアロンのアプリケーションとして実行可能です。

注: 以下の指示を実行する前に、HATS リッチ・クライアント・プラグイン・プロジェクトがすでに作成され、ローカル・テスト環境で実行されているものとします。

クライアント・パッケージを作成するには、以下のステップを実行します。

1. 「**HATS Projects**」ビューで、HATS リッチ・クライアント・プロジェクトを右クリックし、ポップアップ・メニューから「**新規 HATS**」>「**製品構成**」を選択します。これにより、「新規製品構成」ウィザードが起動します。また、「**ファイル**」>「**新規**」>「**その他**」>「**プラグイン開発**」>「**製品構成**」>「**次へ**」を選択して、このウィザードを Rational SDP メニュー・バーから起動することもできます。
2. **com.ibm.hats.rcp.runtime.extension** プラグイン・プロジェクトのフォルダーを親フォルダーとして選択します。
3. 「ファイル名」フィールドに `hostaccess.product` と入力します。
4. 「**起動構成の使用**」ラジオ・ボタンを選択して、**hostaccess** を選択します。

注: この例では、hostaccess は、ローカル・テスト環境でアプリケーションのテストを行ったときに使用した「実行」または「デバッグ」の構成のデフォルト名です。

5. 「**完了**」をクリックします。
6. 「**概要**」タブをクリックし、「**Eclipse 製品のエクスポート・ウィザード**」のリンクをクリックします。
7. 「宛先」の下の「ディレクトリー」フィールドに、Eclipse 製品のエクスポート先ディレクトリーの名前を入力します。例えば、`c:¥hostaccess` と入力します。あるいは、アーカイブ (.zip) ファイルにエクスポートすることもできます。これは、クライアント・パッケージを 1 つのファイル・ダウンロードとして使用できるようにする場合に便利です。
8. 「エクスポート・オプション」で、「**メタデータ・リポジトリーの生成 (Generate metadata repository)**」をクリアします。
9. 「**完了**」をクリックします。

完全な Eclipse クライアント環境が作成されます。アプリケーションをテストするには、Windows Explorer を使用して c:¥hostaccess ディレクトリーにナビゲートします。eclipse フォルダーで、**eclipse.exe** をダブルクリックします。このディレクトリーは、zip して個々のクライアント・マシンにダウンロードできます。クライアント・システムでは、互換性のある Java ランタイム環境 (JRE) 以外のものは不要です。

既存の Eclipse クライアントのパッケージ化 (Eclipse RCP、Lotus Notes、または Lotus Expeditor Client)

HATS リッチ・クライアント・アプリケーションを既存の Eclipse クライアント・ランタイム環境に展開するには、以下の手順を実行する必要があります。

- プロジェクトを Eclipse フィーチャーとしてエクスポート
- HATS ランタイム・フィーチャーのエクスポート
- 更新サイトの作成
- ランタイム環境でのアプリケーションのインストール

以下のセクションでは、これらのタスクのそれぞれを実行する方法を説明しています。

プロジェクトを Eclipse 機能としてエクスポート

HATS リッチ・クライアント・アプリケーション・プラグインをランタイム環境に展開するには、最初にそれを Eclipse フィーチャーにパッケージ化する必要があります。 Eclipse フィーチャーは、関連したプラグインをパッケージ化するために使用するデプロイメント成果物です。プラグインを Eclipse フィーチャーにパッケージ化するには、以下のステップを実行します。

1. 空のディレクトリー (例えば、c:¥myHATS) をローカル・ハード・ディスク上に作成し、エクスポートした HATS アプリケーション・フィーチャーを保管します。 HATS ランタイム・フィーチャーおよび更新サイトの定義を含む site.xml ファイルを保管するには、この同じディレクトリーを使用します。詳しくは、83 ページの『HATS ランタイム・フィーチャーのエクスポート』および 84 ページの『更新サイトの作成』を参照してください。

2. 「**HATS Projects**」ビューからリッチ・クライアント・プロジェクトを右クリックし、ポップアップ・メニューから「**エクスポート**」>「**フィーチャーのエクスポート**」を選択します。

   注:

   ライセンス設定を指定していない場合、このアプリケーションではランタイムが有効にされておらず、実稼働環境では実行できないというメッセージが表示されます。

   HATS リッチ・クライアント・アプリケーションはローカル・テスト環境 (Eclipse、Lotus Notes、または Lotus Expeditor Client) でテストできますが、最初にライセンス設定を指定しなければ、ランタイム (非開発) 環境で実行することはできません。これは、HATS Toolkit 開発環境でアプリケーションを作成し、それを十分にテストできるという意味です。また、アプリケーションをエクスポートし、ユーザーはそれをインストールできますが、アプリケーションに接続することはできません。つまり、ランタイムはロックダウンされ、どんな接続も許可しません。ライセンス設定の指定については、*「HATS スタートアップ・ガイド」* の『HATS ランタイムおよびライセンス設定を有効にする』を参照してください。

   ライセンスを購入済みの場合は、「**ランタイムの使用可能化**」をクリックして、ランタイム・アプリケーションを使用可能にします。まだ購入していない場合は、「**続行**」をクリックしてエクスポート・プロセスを続行します。
3. プロジェクトがまだフィーチャーの一部ではないと想定して、ダイアログはフィーチャーをすぐに作成するように要求を出します。「**はい**」をクリックして、「HATS フィーチャー・プロジェクトを作成」ウィザードを起動します。
4. 「HATS フィーチャー・プロジェクトを作成」ウィザードで、「**フィーチャー ID**」、「**名前**」、「**バージョン**」、および「**プロバイダー**」フィールドを完成させます。

   注: 「**フィーチャー ID**」フィールドを入力すると、「**名前**」フィールドは自動的に入力されます。「**フィーチャー ID**」および「**名前**」フィールドを入力すると、これらのフィールドに対して妥当性検査が実行されます。「**フィーチャー ID**」として入力する値は、作成されるフィーチャー・プロジェクトに使用される名前でもあり、すべての値はプロジェクトの feature.xml ファイルに保管されます。HATS は、「**フィーチャー ID**」フィールド、「**名前**」フィールド、およびバージョンのデフォルトとして名前 `hostaccess` を提供し、「**バージョン**」フィールドのデフォルトとして `1.0.0` を提供します。
5. プラグイン・リストには、ワークスペース内のすべての HATS リッチ・クライアント・プラグイン・プロジェクトからデータが取り込まれます。プラグインを選択して新規フィーチャーに組み込みます。これらのプラグインは、フィーチャーのインストール時にクライアントにインストールされます。「**完了**」をクリックします。
6. 次に、Eclipse の「エクスポート」ウィザードが表示されます。「デプロイ可能なフィーチャー」パネルで、フィーチャーを選択します。

7. 「**宛先**」タブで、「**ディレクトリー**」を選択して、「**参照**」をクリックし、エクスポート・ディレクトリー (例えば、c:¥myHATS) を見つけて選択します。
8. 「**オプション**」タブで、「**個々の JAR アーカイブとしてパッケージ (JNLP およびサイトの更新で必要) (Package as individual JAR archives (required for JNLP and update sites)**」が選択されていることを確認します。その他のオプションは、デフォルトの選択のままでかまいません。
9. 「**完了**」をクリックします。

エクスポート済みの HATS リッチ・クライアント・アプリケーション・フィーチャーは、ランタイム拡張プラグインを自動的に組み込み、HATS ランタイム・フィーチャーへの依存関係を自動的に組み込みます。フィーチャー・プロジェクトは「HATS Projects」ビューには表示されませんが、「ナビゲーター」ビューには表示されます。

HATS リッチ・クライアント・アプリケーション・フィーチャーの向上した拡張構成を実行する場合、「HATS フィーチャー・プロジェクトを作成」ウィザードを使用する代わりに、Eclipse によって提供される標準の Eclipse「新規フィーチャー」ウィザードを使用できます。このウィザードを起動するには、Rational SDP ツールバーから「**ファイル**」>「**新規**」>「**その他**」>「**プラグイン開発**」>「**フィーチャー・プロジェクト**」を選択します。このウィザードは基本的に、「HATS フィーチャー・プロジェクトを作成」ウィザードと同じ機能を実行します。ただし、ランタイム拡張プラグインを自動的に組み込んだり、HATS ランタイム・フィーチャーへの必要な依存関係を組み込んだりすることはありません。

HATS ランタイム・フィーチャーのエクスポート

実稼働環境では、HATS リッチ・クライアント・アプリケーション・フィーチャーは、クライアントにインストールされている HATS ランタイム・フィーチャーに依存します。 HATS リッチ・クライアント・アプリケーション・フィーチャーを「HATS フィーチャー・プロジェクトを作成」ウィザードを使用して作成すると、HATS ランタイム・フィーチャーに対する依存関係が自動的に組み込まれます。この依存関係は、HATS リッチ・クライアント・アプリケーション・フィーチャーがインストールまたは更新されるときに、フィーチャーのインストールまたは更新プロセスが HATS ランタイム・フィーチャーを更新サイトから自動的にプルダウンするというものです。 HATS ランタイム・フィーチャーには以下のフィーチャーおよびプラグインが含まれています。

- フィーチャー: HATS Core (com.ibm.hats.core)
  - プラグイン:
    - com.ibm.hats.core
    - com.ibm.hats.rcp.runtime
    - com.ibm.hats.rcp.ui
    - com.ibm.rcp.doc
    - com.ibm.hostsim
- フィーチャー: SWT 端末 (com.ibm.etools.terminal.beans)
  - プラグイン:
    - com.ibm.eNetwork.beans.HOD
    - com.ibm.etools.hasslite

    - com.ibm.etools.terminal.beans
    - com.ibm.editors.swtbidiextension
- フィーチャー: ホスト画面レンダリング (com.ibm.hsrendering)
  – プラグイン:
    - com.ibm.hsrendering
- フィーチャー: WebFacing Common (com.ibm.iseries.wfcommon)
  – プラグイン:
    - com.ibm.etools.iseries.wfcommon
- フィーチャー: EMF サービス・データ・オブジェクト (SDO) (org.eclipse.emf.ecore.sdo)
  – プラグイン:
    - org.eclipse.emf.ecore.sdo
    - org.eclipse.emf.commonj.sdo

HATS ランタイム・フィーチャーをエクスポートするには、以下のステップを実行します。

1. 「**HATS Projects**」ビューで、HATS リッチ・クライアント・プロジェクトを右クリックし、ポップアップ・メニューから「**エクスポート**」>「**HATS Runtime のフィーチャーをエクスポート**」を選択します。これにより、「HATS Runtime のフィーチャーをエクスポート」ウィザードが起動します。また、「**ファイル**」>「**エクスポート**」>「**HATS**」>「**HATS Runtime のフィーチャーをエクスポート**」を選択して、このウィザードを Rational SDP メニュー・バーから起動することもできます。
2. 「HATS Runtime のフィーチャーをエクスポート」パネルで、「**次の項目としてデプロイ**」に「**ディレクトリー**」を選択し、「**参照**」をクリックして、エクスポート・ディレクトリー (例えば、c:¥myHATS) を見つけて選択します。

   注: 「HATS Runtime のフィーチャーをエクスポート」ウィザードでは、HATS ランタイム・フィーチャーを保存するためのアーカイブ・ファイルまたはディレクトリーを、ローカル・ファイル・システム上で指定することができます。「**アーカイブ**」ファイルへのエクスポートを選択する場合、すべてのフィーチャーとプラグインが単一の .zip アーカイブにパッケージ化されます。 .zip アーカイブ内のルート・ディレクトリー構造は 2 つのサブディレクトリーで始まり、1 つは features、もう 1 つは plugins という名前が付けられています。「**ディレクトリー**」へのエクスポートを選択する場合、フィーチャーおよびプラグインは選択したディレクトリーにエクスポートされます。フィーチャーは features というサブディレクトリーに置かれ、プラグインは plugins というサブディレクトリーに置かれます。アーカイブを Web サーバーの更新サイト・ディレクトリーにアップロードする計画の場合は、「**ディレクトリー**」を選択します。
3. 「**完了**」をクリックします。

更新サイトの作成

HATS アプリケーション・フィーチャーと HATS ランタイム・フィーチャーの両方をエクスポートした後、アプリケーションを展開するために、フィーチャーのイン

ストールおよび更新プロセスで使用される更新サイトを作成する必要があります。更新サイトを作成するには、以下のステップを実行します。

1. 「**HATS Projects**」ビューで、HATS リッチ・クライアント・プロジェクトを右クリックし、ポップアップ・メニューから「**新規 HATS**」>「**更新サイト・プロジェクト**」を選択します。これにより、「新規更新サイト」ウィザードが起動します。また、「**ファイル**」>「**新規**」>「**その他**」>「**プラグイン開発**」>「**更新サイト・プロジェクト**」>「**次へ**」 を選択して、このウィザードを Rational SDP メニュー・バーから起動することもできます。
2. 「更新サイト・プロジェクト」パネルで、以下のようにします。
   a. プロジェクトに名前を付けます (例えば、myHATSUpdateSite)。
   b. 「**デフォルト・ロケーションの使用**」チェック・ボックスのチェックを外し、「**参照**」をクリックして、HATS アプリケーション・フィーチャーおよび HATS ランタイム・フィーチャーをエクスポートしたディレクトリー (例えば、c:¥myHATS) を見つけて選択します。
   c. 「**完了**」をクリックします。
3. 「**サイト・マップ**」タブの site.xml エディター・ビューで、「**新規カテゴリー**」をクリックします。
4. 「カテゴリー・プロパティー」セクションで、以下のようにします。
   a. 「名前」を入力します (例えば、HATS)。
   b. 「ラベル」を入力します (例えば、HATS)。
   c. オプションで「説明」を入力します。
   d. Rational SDP ツールバーの「**保存**」アイコンをクリックします。
5. 「**サイト・マップ**」タブの site.xml エディター・ビューで、「**フィーチャーの追加**」をクリックします。
6. 「フィーチャー選択」パネルで、スクロールダウンして、ご使用の HATS アプリケーション・フィーチャーを選択し (例えば、hostaccess (nnnn)。nnnn はフィーチャーのバージョン番号を表します)、「**OK**」をクリックします。バージョン番号の詳細については、92 ページの『プラグインおよびフィーチャーのバージョン番号の更新』を参照してください。

   注: 「フィーチャーの選択」フィールドにフィーチャー名の入力を始める場合、入力するデータはワイルドカードの働きをし、リスト・ボックス内のフィーチャーのリストは、希望するフィーチャーが表示されるまで絞り込まれます。
7. 「**フィーチャーの追加**」をクリックして、HATS ランタイム・フィーチャーのそれぞれを追加します。
   - com.ibm.etools.terminal.beans (nnnn)
   - com.ibm.hats.core (nnnn)
   - com.ibm.hsrendering (nnnn)
   - com.ibm.iseries.wfcommon (nnnn)
   - org.eclipse.emf.ecore.sdo (nnnn)
8. 「**すべてビルド**」をクリックして、更新サイト site.xml ファイルのビルドを完了します。

注: 「コンパイル・エラーがビルド中に発生しました。ログは、サイト・プロジェクトのルートにある 'logs.zip' で見ることができます。」 というエラー・メッセージが表示されたときは、logs.zip ファイルを確認してください。このファイルに警告のみが含まれている場合は、無視して安全であり、続行できます。

更新サイトが作成されました (この例では c:¥myHATS)。これには、どのフィーチャーおよびそれらのフィーチャーのどのバージョンを、クライアントがダウンロードするためのサイトから使用できるかを示す site.xml ファイルが含まれます。これで、クライアントからアクセス可能な Web サーバーに、この更新サイト・ディレクトリーを移動できます。 クライアント・システムで更新サイトを定義し、HATS リッチ・クライアント・アプリケーションをランタイム環境にインストールする方法については、『ランタイム環境でのアプリケーションのインストール』を参照してください。

注: HATS サービスの更新をインストールした後、変更を有効にするには、HATS ランタイム・フィーチャーを再エクスポートして再デプロイする必要があります。更新サイトの site.xml ファイルを、バージョン番号の変更を反映するように変更する必要があります。バージョンは、「HATS Runtime のフィーチャーをエクスポート」ウィザードの最初のパネルにリストされます。詳しくは、 92 ページの『プラグインおよびフィーチャーのバージョン番号の更新』を参照してください。

ランタイム環境でのアプリケーションのインストール

HATS リッチ・クライアント・アプリケーションは、Eclipse RCP 環境、Lotus Notes 環境、および Lotus Expeditor Client 環境で実行できます。以下のセクションでは、これらの環境のそれぞれに HATS リッチ・クライアント・アプリケーションをインストールする方法を説明します。

Eclipse RCP:

HATS リッチ・クライアント・アプリケーションを Eclipse リッチ・クライアント環境にダウンロードおよびインストールするには、そのアプリケーションを含む更新サイトにアクセスするようにクライアント環境を構成する必要があります。デフォルトでは、自動インストールおよび更新は使用不可にされているため、ユーザーまたは管理者が標準の Eclipse Software Update 機能を使用して新規のフィーチャーまたは更新を手動でインストールする必要があります。これを実行するには、クライアント・システムで以下のステップを実行します。

注: 以下の説明は、Eclipse 3.6 以降を対象にしています。それより前のバージョンの Eclipse を使用している場合は、適切な Eclipse 資料に従ってフィーチャーをインストールしてください。

1. Eclipse クライアントのメニュー・バーで、「**ヘルプ**」>「**新規ソフトウェアのインストール (Install New Software)**」 を選択します。

   注: Eclipse クライアントの構成によっては、このメニュー項目が別のメニューの下に配置されています。
2. 「**追加**」ボタンをクリックしてから、「**ローカル**」ボタンをクリックします。

3. HATS プロジェクト用に作成した更新サイトの場所まで参照して行き、「**OK**」をクリックします。
4. HATS 更新サイトを選択します。更新サイトに HATS という名前のカテゴリーを割り当てた場合には、「**HATS**」の横のチェック・ボックスを選択して、その下のすべてのチェック・ボックスも同様に選択されていることを確認します。または、「**全選択**」ボタンをクリックすることもできます。更新サイトのカテゴリーがリストされていない場合は、「**項目をカテゴリー別にグループ化 (Group items by category)**」チェック・ボックスが選択されていることを確認してください。「**次へ**」をクリックします。
5. インストールされる項目を検討し、確認します。「**次へ**」をクリックします。
6. 「**使用条件の条項に同意します**」を選択し、「**終了**」をクリックします。
7. インストール中のソフトウェアに署名なしのコンテンツが含まれていることを示すセキュリティー警告が表示された場合は、「**OK**」をクリックします。
8. HATS アプリケーションがインストールされました。
9. ワークベンチを再始動するプロンプトが表示されたら、「**はい**」をクリックします。
10. これで、まずホスト・アクセス・パースペクティブを開くことによりアプリケーションを実行できます (「**ウィンドウ**」->「**パースペクティブを開く**」->「**ホスト・アクセス**」)。ご使用の HATS アプリケーションが「アプリケーション」ビューにリストされます。アプリケーションを開始するには、アプリケーションをダブルクリックするか、または右クリックして「**オープン**」を選択します。

Eclipse RCP 環境の詳細については、http://www.eclipse.org/home/categories/rcp.php を参照してください。

Lotus Notes:

Lotus Notes 環境では、フィーチャーおよびプラグインとして提供されるアプリケーションのインストールにウィジェットが使用されます。ウィジェットを後でウィジェット・カタログに公開して、その他で使用できるようにすることができます。ウィジェットの詳細については、Lotus Notes Knowledge Center の『ウィジェットおよび Live Text』のセクション (http://publib.boulder.ibm.com/infocenter/domhelp/v8r0/index.jsp?topic=/com.ibm.notes85.help.doc/wid_app_overview_c.html) を参照してください。

HATS リッチ・クライアント・アプリケーションを Lotus Notes 環境にダウンロードおよびインストールするには、クライアント・システムで以下のステップを実行します。

1. Lotus Notes クライアントを開始し、「**ファイル**」>「**プリファレンス**」>「**ウィジェット**」を選択します。
2. 「**Widget ツールバーと「My Widgets サイドバー」パネルを表示 (Show Widget Toolbar and the My Widgets Sidebar panel)**」を選択します。「**OK**」をクリックします。

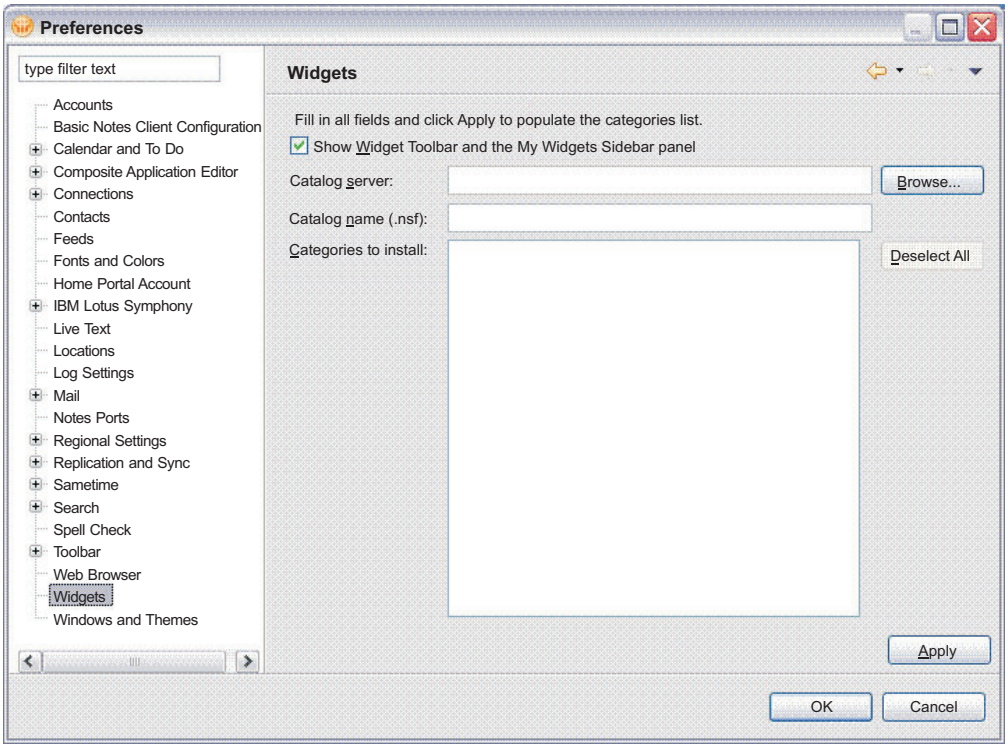


図 7. Lotus Notes ウィジェットのプリファレンス

3. Widget ツールバーは次のようになります。

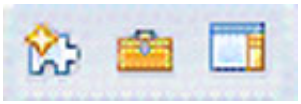


図 8. Lotus Notes Widget ツールバー

4. 「My Widgets サイドバー」パネルは、次のようになります。

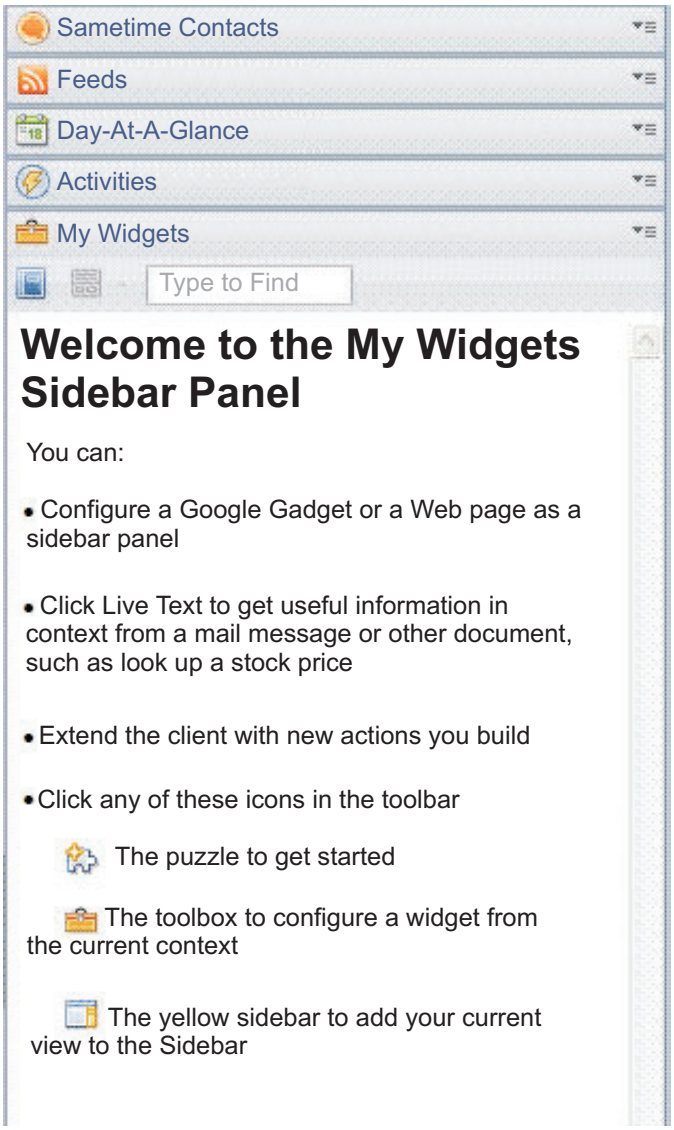


図 9. My Widgets サイドバー

5. Widget ツールバーから「**ウィジェット入門 (Getting Started with Widgets)**」を選択します。
6. 「**フィーチャーとプラグイン (Features and Plugins)**」を選択してから、「**次へ**」を選択します。

7. 「**URL を入力 (Enter the URL)**」フィールドに、次のフォーマット (file:/// で始まる必要がある) を例として使用して、更新サイトの site.xml ファイルへのパスを入力します: file:///c:¥hatsupdatesite¥site.xml。「**ロード**」ボタンをクリックします。
8. 「**インストールするフィーチャー (Features to Install)**」で、すべてのフィーチャーを選択し、「**次へ**」を選択します。
9. オプション: 「**ウィジェット名 (Widget name)**」フィールドに、HATS Provisioning Widget または別の適切な名前を入力できます。
10. オプション: ウィジェットにイメージを関連付ける必要がある場合は、必要なイメージを更新サイトのフォルダーにコピーします。次に、「**イメージ URL (Image URL)**」フィールドに、次のフォーマット (file:/// で始まる必要がある) を例として使用して、イメージへのパスを入力します: file:///c:¥hatsupdatesite¥myhatsimage.png。
11. 要求されたら、プラグインをインストールすることを選択します。

「My Widgets サイドバー」パネルは、次のようになります。

図 10. 例として HATS Provisioning Widget が指定された My Widgets サイドバー

12. 要求されたら、Lotus Notes を再始動します。

「**オープン**」>「**ホスト・アクセス**」を選択することにより、アプリケーションを実行できます。ご使用の HATS アプリケーションが「**アプリケーション**」ビューにリストされます。アプリケーションを開始するには、アプリケーションをダブルクリックするか、または右クリックして「**オープン**」を選択します。

HATS アプリケーションをアンインストールするには、HATS Provisioning Widget (または、それに指定した任意の名前) を右クリックし、「**除去**」を選択します。要求されたら、Lotus Notes を再始動します。

Lotus Expeditor Client:

Lotus Expeditor Client 環境では、アプリケーションの管理および展開を以下のいくつかの方法で制御できます。

- Lotus Expeditor Server。Device Manager Server コンポーネントが含まれています。
- WebSphere Portal。アプリケーションに対して、集中管理された役割ベースのアクセスが可能です。
- Eclipse 更新サイト。標準の Web サーバーを使用して、プロビジョニングおよび更新を提供します。

Lotus Expeditor Client では、デフォルト・インストールに以下のフィーチャーを追加する必要があります。

- Lotus Expeditor Desktop 開発ユーティリティーLotus
- コア JVM フィーチャー - J2SE (Lotus Expeditor Client バージョン 6.2.0 以前を使用する場合のみ)

Lotus Expeditor Client で、「**ファイル**」>「**アプリケーション**」>「**インストール**」>「**インストールする新しいフィーチャーの検索**」>「**フォルダー・ロケーションの追加**」を選択します。Lotus Expeditor Client 製品ファイルの desktop¥updates¥platform フォルダーまで参照して行き、「**終了**」を選択します。「**updates/platform**」および「**Development**」を展開します。「**Lotus Expeditor デスクトップ開発ユーティリティー**」の横のチェック・マークにマークを付けます。「**次へ**」を選択し、ウィザードを完了します。

Lotus Expeditor Client バージョン 6.2.0 以前の場合、コア JVM フィーチャー - J2SE をインストールするには、Lotus Expeditor Client で、「**ファイル**」>「**アプリケーション**」>「**インストール**」>「**インストールする新しいフィーチャーの検索**」>「**フォルダー・ロケーションの追加**」を選択します。Desktop Runtime Environment サイトの場所 (update.site.dre.client ディレクトリー) まで参照して行き、「**OK**」を選択します。デフォルトのサイト名を受け入れ、「**OK**」および「**終了**」を選択します。このサイトおよび「**ランタイム・コンポーネント (Runtime Components)**」を展開します。「**Core JVM フィーチャー -JSE (Core JVM Feature - J2SE)**」の横のチェック・マークにマークを付けます。「**次へ**」を選択し、ウィザードを完了します。

詳しくは、Lotus Expeditor Knowledge Center (http://publib.boulder.ibm.com/infocenter/ledoc/v6r2/index.jsp) の『Lotus Expeditor アプリケーションのアセンブルおよびデプロイ』セクションを参照してください。

HATS リッチ・クライアント・アプリケーションを Lotus Expeditor Client 環境にダウンロードおよびインストールするには、そのアプリケーションを含む更新サイトにアクセスするようにクライアントを構成する必要があります。これを実行するには、クライアント・システムで以下のステップを実行します。

1. Lotus Expeditor Client のメインメニューで、「**ファイル**」>「**アプリケーション**」>「**インストール**」を選択します。
2. 「フィーチャーの更新」パネルで、「**インストールする新規フィーチャーを検索**」を選択し、「**次へ**」をクリックします。
3. 「アプリケーションのロケーション」パネルで、「**リモート・ロケーションの追加**」または「**フォルダー・ロケーションの追加**」のうち該当する項目をクリックします。
   - 「リモート・ロケーションの追加」の場合は、「新しい更新サイト」パネルで、更新サイトの名前および更新サイトをホスティングする Web サーバーの URL を入力して、「**OK**」をクリックします。
   - 「フォルダー・ロケーションの追加」の場合は、「フォルダーの参照」パネルで更新サイトを格納しているフォルダーを参照し、「**OK**」をクリックします。
4. 「アプリケーションのロケーション」パネルで、HATS 更新サイトを選択し、「**完了**」をクリックします。
5. 「検索結果」パネルで、インストールするフィーチャーを選択し、「**次へ**」をクリックします。
6. 「フィーチャー・ライセンス」パネルで、「**使用条件の条項に同意します**」 を選択し、「**次へ**」をクリックします。
7. 「インストール」パネルで、「**完了**」をクリックします。
8. 要求されたら、プラグインをインストールすることを選択します。
9. HATS アプリケーションがインストールされました。
10. ワークベンチを再始動するプロンプトが表示されたら、「**はい**」をクリックします。
11. これで、まずホスト・アクセスを開くことによりアプリケーションを実行できます (「**開く**」->「**ホスト・アクセス**」)。ご使用の HATS アプリケーションが「アプリケーション」ビューにリストされます。アプリケーションを開始するには、アプリケーションをダブルクリックするか、または右クリックして「**オープン**」を選択します。

自動更新処理:

リッチ・クライアント環境により、インストールされたアプリケーションの自動更新が可能になります。サーバーの更新が検査される頻度を制御するために、プリファレンスを使用できます。更新をアプリケーションにアップロードしたら、リッチ・クライアントは更新をダウンロードしてインストールします。新規更新のみがダウンロードされてインストールされたことを確認するために、バージョン検査が実行されます。たいていの場合、ユーザーは環境を再始動して、更新されたアプリケーションをアクティブ化する必要があります。詳細については、適切な Eclipse RCP、Lotus Notes、または Lotus Expeditor の資料を参照してください。

プラグインおよびフィーチャーのバージョン番号の更新

すでにインストールされているプラグインを更新するには、その前にプラグインのバージョンを更新する必要があります。これは Eclipse が、そのバージョンを使用してプラグインが変更されたかどうかを判別するためです。例えば、プロジェクト

内で画面カスタマイズを追加するか、またはマクロを変更する場合、プロジェクトを再エクスポートする前にプラグインのバージョンを更新する必要があります。

プラグインのバージョンを更新するには、そのプロジェクトの「プロジェクト設定」エディターで、「**概要**」タブの「**プラグイン・マニフェストをオープンしてください。**」リンクをクリックします。「**概要**」タブの「一般情報」の下で、バージョンを更新します。Eclipse バージョンの形式は `major.minor.service` です。プロジェクトに対して小さい変更を行う場合は、バージョンの `service` セグメントを更新します。例えば、プラグインを二度目にエクスポートする前に、プラグインのバージョンを `1.0.1` に設定します。

「ライセンス設定」ウィザードは、ランタイム拡張プロジェクト (com.ibm.hats.rcp.runtime.extension) を更新するため、ウィザードを実行した後、このプラグインのバージョンを更新する必要があります。これを行うには、「ナビゲーター」ビューで、プロジェクトのルートにある plugin.xml ファイルを開きます。「**概要**」タブの「一般情報」の下で、バージョンを更新します。

また、フィーチャーはプラグインで構成されている (さらに、フィーチャーにはバージョンもある) ため、組み込まれたプラグインのいずれかのバージョンを更新する場合は、必ずフィーチャーのバージョンも更新する必要があります。これを行うには、「ナビゲーター」ビューに切り替え、フィーチャー・プロジェクト内にある feature.xml ファイルを開きます。デフォルトでは、フィーチャー・プロジェクトの名前は `hostaccess` です。「**概要**」タブの「一般情報」の下で、バージョンを更新します。また、フィーチャーがプラグインの新規バージョンを正しくリストしていることを確認してください。それには、「**プラグイン**」タブの「**バージョン**」ボタンをクリックし、「**プラグインとフラグメント・マニフェストからバージョンをコピー**」を選択します。

フィーチャーのバージョンを更新した後、フィーチャーの新規バージョンがリストされることを確認するために、site.xml ファイルを更新する必要があります (該当する場合)。

注: アプリケーションをローカル・テスト環境で実行またはデバッグする場合は、このステップは必要ありません。

HATS リッチ・クライアント・アプリケーションの管理

HATS リッチ・クライアント・アプリケーションの場合、ログ設定やトレース設定などのランタイム設定は、ランタイム拡張プラグイン (com.ibm.hats.rcp.runtime.extension) の 2 つのファイルで指定されます。

- runtime.properties ファイルは、ランタイム環境および「実行」モードで稼働するローカル・テスト環境で、HATS によって使用される設定を指定するために使用されます。
- runtime-debug.properties ファイルは、「デバッグ」モードで稼働するローカル・テスト環境で使用されるランタイム設定を指定するために使用されます。

さまざまなテスト・モードでの実行方法の詳細については、「*HATS スタートアップ・ガイド*」の『リッチ・クライアント・プロジェクトのテスト・モード』を参照してください。

HATS Toolkit で、これらのファイルを編集して、設定 (トレースやトレース・レベルの設定など) を変更できます。ファイルは、ランタイム拡張プラグイン・プロジェクトのルートにあります。HATS Toolkit で runtime.properties ファイルが変更された場合、その変更は保持され、HATS アプリケーションをランタイム環境に展開するときに有効になります。ランタイム環境では、runtime.properties ファイルがランタイム拡張プラグイン・ディレクトリー (例えば、*<eclipse_install_dir>*/eclipse/plugins/com.ibm.hats.rcp.runtime.extension_1.0.0) にあります。 runtime-debug.properties ファイルは、ランタイム環境では無視されます。これらのファイルでのログ設定とトレース設定について詳しくは、557 ページの『付録 A. ランタイム・プロパティー・ファイル』を参照してください。

ログ設定とトレース設定を構成し、ランタイム環境で問題判別データを収集する方法について詳しくは、99 ページの『プリファレンス』を参照してください。

ワークスペース内には、1 つのランタイム拡張プラグイン・プロジェクトがあり、ランタイム環境にはプラグインの 1 つのバージョンだけがあります。そのため、単一の HATS リッチ・クライアント・ランタイム環境では、ログおよびトレース設定は、すべての HATS リッチ・クライアント・アプリケーションで同じです。

ライセンス設定は、ランタイム拡張プラグイン (com.ibm.hats.rcp.runtime.extension) の runtime.properties ファイルにも保管されます。そのため、ライセンスのトラッキングは、個々の HATS リッチ・クライアント・アプリケーションのレベルではなく、リッチ・クライアント環境レベルで実行されます。ユーザーが購入済みのライセンス数を超えた場合、ログに警告メッセージが表示されます。ライセンス設定の指定については、*「HATS スタートアップ・ガイド」*の『HATS ランタイムおよびライセンス設定を有効にする』を参照してください。

ランタイム環境

「アプリケーション」ビュー

「アプリケーション」ビューは、ランタイム環境にインストールされたすべてのアクティブな HATS リッチ・クライアント・アプリケーションをユーザーに表示します。ビューはアプリケーション表示名でソートされます。デフォルトでは、このビューは、ユーザーが初めて「ホスト・アクセス」パースペクティブを開くときに表示されます。ユーザーは、「アプリケーション」ビューでアプリケーションをダブルクリックするか、またはアプリケーション名を右クリックして「**オープン**」を選択することにより、HATS アプリケーションの新規インスタンスを開始します。

HATS アプリケーションの新規インスタンスが開始すると、その「変換」ビューが表示されます。 詳細については、97 ページの『「変換」ビュー』を参照してください。デフォルトでは、新規にオープンした「変換」ビューの名前がアプリケーションの名前になります。アプリケーションの複数のインスタンスが開始される場合、「変換」ビューの名前は、アプリケーションの名前とセッションのインスタンスを示す番号の組み合わせになります。例えば、`Order Entry` という名前のアプリケーションの 2 番目のインスタンスのタイトルは、`Order entry : 2` となります。

デフォルトでは、ユーザーは同じアプリケーションの複数インスタンスを起動できます。複数インスタンスの起動を使用不可にするように plugin.xml ファイルを変更

できます。また、「アプリケーション」ビューを表示しないで、代わりにユーザーが使用可能なビューを制御するように決定することもできます。これにより、アプリケーションのインスタンスの数も制限できます。詳しくは、「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

アプリケーションがローカル・テスト環境で実行されており、表示端末を表示することを選択した場合、システムに対する接続が行われると、「端末の表示」ウィンドウが表示されます。セッションが終了するとき、またはクライアントがシャットダウンされる場合、このウィンドウはクローズします。

ポップアップ・メニュー

ポップアップ・メニューは、ユーザーがビュー内の 1 つ以上のアプリケーションに対してアクションを実行できるように提供されます。メニューを表示するには、1 つ以上のアプリケーション名を選択し、右クリックします。以下のメニュー項目を使用できます。

オープン

このメニュー項目を選択すると、選択されたアプリケーションの新規インスタンスが開始されます。複数のアプリケーションが選択される場合、選択された各アプリケーションのインスタンスが開始されます。

印刷ジョブを表示

このメニュー項目は、少なくとも 1 つの印刷対応アプリケーションが選択される場合にのみ使用可能になります。このメニュー項目を選択すると、「印刷ジョブ (3270)」または「印刷ジョブ (5250)」ビューのいずれかが開きます。このビューには、キューに入れられた印刷ジョブのリストが表示されます。詳細については、417 ページの『第 15 章 印刷サポートの使用可能化』を参照してください。

プロパティー

このメニュー項目は、1 つのアプリケーションだけが選択された場合に使用可能になります。このメニュー項目を選択すると、ユーザーはそのアプリケーションに関する情報を表示するだけでなく、アプリケーションに関連した設定を構成することができます。以下のページは、選択したアプリケーションの「プロパティー」ダイアログで提供されます。

情報
「情報」ページには、選択したアプリケーションに関する情報が表示されます。これには、その名前、ID、バージョン、およびプラグインの物理的な場所が含まれます。また、アプリケーションによって使用されている HATS ランタイム・プラグインのバージョンも示されます。この情報は、ユーザーが環境に関する情報をサポート・チームと通信する必要がある場合に役立ちます。ユーザーはこのページの値を変更できません。これは通知目的のみです。

接続パラメーター

ユーザーが接続パラメーターをオーバーライドできないようにすることを選択した場合、この項目はリストに表示されません。これがデフォルトです。詳細については、130 ページの『接続パラメーターのオーバーライド』を参照してください。

オーバーライドを許可する場合、このページでユーザーは接続パラメーターをオーバーライドすることができます。この機能は Web

アプリケーション用の HATS 機能に相当します。この HATS 機能は、最初にアプリケーションにアクセスするときに URL を使用して接続パラメーターをオーバーライドできるようにします。接続パラメーターのオーバーライドを使用する例は、ユーザーがワークステーション ID または LU 名をオーバーライドする必要がある場合です。

ユーザーがオーバーライドを設定するには、「**追加**」ボタンをクリックし、「**名前**」フィールドのドロップダウン・リストからパラメーターを選択して、「**値**」フィールドに値を設定します。提供された接続パラメーター名をオーバーライドできるかどうかをユーザーに警告するために、ダイアログ内の妥当性検査は実行されません。デフォルトでは、最初に追加されるときにオーバーライド設定が使用可能になります。パラメーター名の隣のチェック・ボックスをクリアすると、この設定を使用不可にすることができます。

ユーザーが接続パラメーターをオーバーライドすると、設定はローカル・ワークスペースに保存されます。セッションのインスタンスが開始されるときに、オーバーライドされたパラメーター値が使用されます。

接続パラメーターのオーバーライドを使用してリッチ・クライアント・アプリケーションのインスタンスをプログラマチックに開始する方法については、「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

変数 ユーザーがグローバル変数をオーバーライドできないようにすることを選択した場合、この項目はリストに表示されません。これがデフォルトです。詳細については、141 ページの『グローバル変数のオーバーライド』を参照してください。

オーバーライドを許可する場合、このページでユーザーはグローバル変数をオーバーライドすることができます。この機能は Web アプリケーション用の HATS 機能に相当します。この HATS 機能は、最初にアプリケーションにアクセスするときに URL を使用してグローバル変数および共用グローバル変数をオーバーライドできるようにします。

この機能は、ユーザー固有の変数データをアプリケーションで使用できるようにします。例えば、グローバル変数の値から提供されたユーザー名およびパスワードを要求するマクロを記録できます。ユーザーは変数のオーバーライド・ページでこれらの値を提供することができます。サインオン画面が認識されるときにマクロが実行されると、これらの値が使用されます。

ユーザーがオーバーライドを設定するには、「**追加**」ボタンをクリックし、「**名前**」フィールドのドロップダウン・リストから変数名を選択して、「**値**」フィールドに値を設定します。さらに、グローバル変数が「**共用**」 かどうかを指定します。提供されたグローバル変数名をオーバーライドできるかどうかをユーザーに警告するために、ダイアログ内の妥当性検査は実行されません。デフォルトでは、最初に追加されるときにオーバーライド設定が使用可能になり

ます。変数名の隣のチェック・ボックスをクリアすると、この設定を使用不可にすることができます。

ユーザーがグローバル変数をオーバーライドすると、設定は環境のデータ・ディレクトリーに保管されます。セッションのインスタンスが開始されるときに、オーバーライドされたグローバル変数値が使用されます。

グローバル変数のオーバーライドを使用してリッチ・クライアント・アプリケーション・インスタンスをプログラマチックに開始する方法については、「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

印刷画面サポート

リッチ・クライアント環境では、ユーザーは、「**ファイル**」 メニューから「**印刷**」を選択することにより、現行の画面変換を印刷できます。デフォルトでは、Eclipse は「印刷」ダイアログを表示します。このダイアログで、ユーザーはプリンターを選択したり、またその他のオプションを構成したりできます。ユーザーが「**OK**」をクリックした後、変換ビューの変換領域が選択したプリンターに送信されます。

注: Windows システムでは、スクロール・バーを使用することによってのみ表示される部分を含む、変換全体が印刷されます。Linux システムでは、表示される変換部分のみが印刷され、スクロールして表示する必要がある非表示部分は印刷されません。

「変換」ビュー

「変換」ビューは、リッチ・クライアント環境内の変換のメイン領域です。それぞれの HATS リッチ・クライアント・プロジェクトには、1 つの変換ビューが関連付けられます。プロジェクトが作成されると、そのプロジェクトの plugin.xml ファイルにビューが登録されます。ビューは以下の 4 つの領域で構成されます。

- ツールバー
- テンプレートおよび変換領域
- ホスト・キーパッド
- OIA 状況域

ツールバー

変換ビュー・ツールバーには、アプリケーション・レベルのアクション用のボタンと、画面レンダリング中にツールバー・ウィジェットによって提供されるアクション用のボタンが含まれます。デフォルトでは、ツールバーが表示され、テキストのみの表題が付いたボタンが表示されます。これらの設定はプロジェクト設定で変更できます。詳細については、121 ページの『ツールバー RCP のみ』を参照してください。

ツールバーは 2 つのセクションに分割されます。ツールバーの右側は、HATS Web アプリケーションのアプリケーション・キーパッドに相当し、アプリケーション・キーパッド・ボタンがあります。表示するボタンの構成方法については、121 ページの『アプリケーション・キーパッド』を参照してください。左側には、ツールバー・ウィジェットによって提供されるボタンが表示されます。詳細については、347 ページの『ツールバー RCP のみ』を参照してください。

ボタンの機能の説明を提供するツールチップ・テキストが、すべてのアプリケーション・キーパッド・ボタンで使用できます。ツールバー・ウィジェットを使用して、追加するボタン上にツールチップ・テキストを設定できます。

ツールバーのアプリケーション・キーパッド・ボタンの上にある「最小化」および「最大化」アイコンの隣が「メニュー」アイコンです。このアイコンをクリックすると、以下のメニュー項目を持つメニューが表示されます。

キーボード・サポートの切り替え
ユーザーはキーボード・サポートを使用可能にしたり、使用不可にしたりできます。

接続の詳細
このアプリケーションのホスト接続の詳細を表示します。

プロパティー
このアプリケーションのプロパティーを表示します。

テンプレートおよび変換領域

テンプレートおよび変換領域は変換ビューのメイン部分で、ここにすべての画面変換が表示されます。また、エラー・メッセージ、切断メッセージ、およびマクロ・ハンドラー・パネルが表示される領域です。

ポップアップ・メニュー
ポップアップ・メニューは、ユーザーが現行アプリケーションに関する情報を表示できるようにする変換領域に提供されます。以下のメニュー項目を使用できます。

プロパティー
この項目を選択すると、現行アプリケーションのプロパティー・ダイアログが開きます。詳細については、95 ページの『ポップアップ・メニュー』を参照してください。

ランタイム状況表示パネル
HATS Web アプリケーションと同様に、HATS リッチ・クライアント・アプリケーションが停止または切断されたり、エラーが検出されたり、あるいは使用中である場合、ユーザーに対してパネルが表示されます。

ホスト・キーパッド

ホスト・キーパッドには、プロジェクト設定で表示することを選択したホスト・キーが表示されます。ホスト・キーパッドの構成方法については、122 ページの『ホスト・キーパッド』を参照してください。デフォルトでは、ホスト・キーパッドは表示されません。

OIA 状況域

OIA 状況域には、プロジェクト設定で構成したセッションの状態に関する情報が表示されます。 OIA 状況域の構成方法については、122 ページの『オペレーター情報域』を参照してください。

ビューのクローズ

ビューがクローズされると、ホスト・システムへの接続がクローズされます。これは、ユーザーがビューを明示的にクローズしたり、リッチ・クライアント環境をク

ローズする場合に発生します。クローズされると、切断および停止イベントで定義されたアクションのセットが処理されます。

注: HATS がまだ要求を処理している間にユーザーがビューをクローズしようとすると (例えば、マクロの再生を開始したボタンをユーザーがクリックした場合)、セッションが使用中であることを示すメッセージが表示されます。ユーザーが続行する場合、セッションは終了され、切断および停止イベントのアクションは実行されない場合があります。

ワークステーション ID のプロンプト表示

ホスト接続を作成または編集するときに、ユーザーに対して 5250 ホストに接続するためのワークステーション ID の入力を求めるプロンプトが表示されるように指定できます。詳細については、157 ページの『接続の作成』を参照してください。ワークステーション ID のプロンプト表示を指定する場合、最初にホストへの接続を行うときに、ユーザーにプロンプト・ダイアログが表示されます。そこで、ユーザーは「**ワークステーション ID**」フィールドにワークステーション ID を入力できます。

選択した場合にワークステーション ID が保存され、今後はプロンプトが表示されないようにするチェック・ボックスが提供されます。チェック・ボックスが選択される場合、後でこのプロンプト表示を復元するには、アプリケーションのプロパティーで workstationID 接続パラメーターのオーバーライドを使用不可にします。詳細については、95 ページの『接続パラメーター』を参照してください。

LU 名のプロンプト表示

ホスト接続を作成または編集するときに、ユーザーに対して 3270E ホストに接続するための LU 名の入力を求めるプロンプトが表示されるように指定できます。詳細については、157 ページの『接続の作成』を参照してください。LU 名のプロンプト表示を指定する場合、最初にホストへの接続を行うときに、ユーザーにプロンプト・ダイアログが表示されます。ユーザーは、「**LU 名**」フィールドに LU 名を入力する必要があります。

選択した場合に LU 名が保存され、今後はプロンプトが表示されないようにするチェック・ボックスが提供されます。チェック・ボックスが選択される場合、後でこのプロンプト表示を復元するには、アプリケーションのプロパティーで LUName 接続パラメーターのオーバーライドを使用不可にします。詳細については、95 ページの『接続パラメーター』を参照してください。

プリファレンス

デフォルトでは、ランタイム・プラットフォームのメニュー・バーから「**ファイル**」>「**プリファレンス**」>「**ホスト・アクセス**」を選択することにより、ユーザーはプリファレンスにアクセスできます。表示されるページは、ランタイム拡張プラグインに登録されます。これらの機能をユーザーに提供しない場合、ランタイム拡張プラグインの plugin.xml ファイル内の該当する宣言を除去できます。この方法について詳しくは、「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」の『HATS ランタイム拡張プラグイン』セクションにある org.eclipse.ui.preferencePages 拡張ポイント ID の説明を参照してください。以下の設定が選択されています。

- 印刷
- トラブルシューティング

印刷プリファレンス

以下の印刷プリファレンスは、3270 印刷をサポートするアプリケーションのユーザーが設定できます。

デフォルト印刷ジョブ・アクション

開始時 印刷ジョブの開始時に、以下のプリファレンスを設定できます。

- 印刷ジョブ・ビューのアクティブ化
- メッセージの表示

  印刷ジョブが開始されていることをユーザーに示すメッセージが表示されます。
- なし

完了時 印刷ジョブの完了時に、以下のプリファレンスを設定できます。

- 印刷ジョブ・ビューのアクティブ化
- メッセージの表示

  「印刷ジョブ {0} が完了しました。」というメッセージが表示されます。ここで、{0} は、印刷ジョブ名です。
- オープン
- なし

エラー時

印刷ジョブの失敗時に、以下のプリファレンスを設定できます。

- 印刷ジョブ・ビューのアクティブ化
- メッセージの表示

  「印刷ジョブ {0} でエラーが発生しました。」というメッセージが表示されます。ここで、{0} は、印刷ジョブ名です。
- 削除
- なし

終了時に印刷ジョブを削除

選択した場合、環境がクローズされると、印刷ジョブは自動的に削除されます。これは、ワークステーションを他のユーザーと共有するユーザーにとって便利です。

トラブルシューティング・プリファレンス

以下のトラブルシューティング・プリファレンスは、ユーザーが設定できます。ログ設定とトレース設定について詳しくは、93 ページの『HATS リッチ・クライアント・アプリケーションの管理』を参照してください。

トレース設定

次の HATS ランタイム・トレースはユーザーが有効にすることができます。

ランタイムのトレースを使用可能にする
アプリケーション内の接続のトレースを使用可能にします。

ウィジェットのトレースを使用可能にする
アプリケーション内のウィジェットのトレースを使用可能にします。

アクションのトレースを使用可能にする
アプリケーション内のイベント・アクションのトレースを使用可能にします。

HOD のトレースを使用可能にする
アプリケーション内の Host On-Demand トレースを使用可能にします。

コンポーネントのトレースを使用可能にする
アプリケーション内のコンポーネントのトレースを使用可能にします。

ユーティリティーのトレースを使用可能にする
アプリケーション内のランタイム・ユーティリティーのトレースを使用可能にします。

マクロのトレースを使用可能にする
マクロ実行をトレースします。システム・パフォーマンスに影響するため、このトレースはマクロのデバッグでのみ使用してください。

表示端末の表示
HATS アプリケーションの実行中に表示端末にホスト・アプリケーションのビューを表示するには、このボックスを選択します。

ホスト・シミュレーションの記録を使用可能にする
ランタイム中にホスト・シミュレーション記録を使用可能にするには、このボックスを選択します。記録は、次にリッチ・クライアント・アプリケーションが開始するときに開始し、ホスト・セッションがクローズされると (例えば、ユーザーが変換ビュー・ツールバーの切断項目をクリックしたとき) 終了します。

トレース・ファイルはリッチ・クライアント環境のホスト・シミュレーション・フォルダーに保存されます。例えば、Windows の Eclipse RCP 環境では、*<Eclipse installation directory>*¥plugins¥*<ApplicationName_version>*¥profiles¥hostsimulations¥ に保存されます。 Windows の Lotus Notes 環境では、*<Lotus Notes installation directory>*¥Data¥workspace¥applications¥eclipse¥plugins¥*<ApplicationName_version>*¥profiles¥hostsimulations¥ に保存されます。 Windows の Lotus Expeditor Client 環境では、*<Lotus Expeditor Client installation directory>*¥shared¥eclipse¥plugins¥*<ApplicationName_version>*¥profiles¥hostsimulations に保存されます。 Linux の Lotus Expeditor Client 環境では、/opt/IBM/Lotus/Expeditor/shared/eclipse/plugins/*<ApplicationName_version>*/profiles/hostsimulations/ に保存されます。

トレース・ファイル名は、ApplicationName_ConnectionName_Date (yyyymmdd)_Time(hhmmss)_Number というテンプレートを使用して指定されます (例えば、MyApplication_main_20060101_134543_1)。

「トレース出力」タブ

ファイル名
各ランタイム環境に固有の一連のトレース・ファイルを生成するためのテンプレートとして使用される名前。デフォルトは **trace.txt** です。

最大ファイル・サイズ
最大のファイル・サイズを指定します (KB 単位)。デフォルトは **10240** です。

ファイルの最大数
トレース・ファイルの最大数を指定します。デフォルトは「**5**」です。

「表示」ボタン
トレース・ファイルがデフォルトのシステム・エディターで開かれるようにするには、このボタンをクリックします。指定されたテンプレート・ファイル名に一致するすべてのトレース・ファイルが単一のファイルに結合されてから開かれます。例えば、trace.txt がテンプレート・ファイル名であり、trace1.txt、trace2.txt、および trace3.txt が存在する場合、それらはすべて単一のファイルに正しい順序で結合されてから、開かれます。デフォルト・エディターが構成されていない場合、エラー・メッセージが表示されます。

「クリア」ボタン
現行のトレース・ファイルをクリアするかどうかを指示するには、このボタンをクリックします。

「メッセージ出力」タブ

ファイル名
各ランタイム環境に固有の一連のログ・ファイルを生成するためのテンプレートとして使用される名前。デフォルトは **messages.txt** です。

最大ファイル・サイズ
最大のファイル・サイズを指定します (KB 単位)。デフォルトは **512** です。

ファイルの最大数
トレース・ファイルの最大数を指定します。デフォルトは **2** です。

「表示」ボタン
現行のログ・ファイルがデフォルトのシステム・エディターで開かれるようにするには、このボタンをクリックします。指定されたテンプレート・ファイル名に一致するすべてのログ・ファイルが単一のファイルに結合されてから、開かれます。例えば、messages.txt がテンプレート・ファイル名であり、messages1.txt、messages2.txt、および messages3.txt が存在する場合、それらはすべて単一のファイル

に正しい順序で結合されてから、開かれます。デフォルト・エディターが構成されていない場合、エラー・メッセージが表示されます。

「クリア」ボタン
現行のログ・ファイルをクリアするかどうかを指示するには、このボタンをクリックします。

「シミュレーション出力」タブ

ファイルの最大数
「ホスト・シミュレーションの記録を使用可能にする」チェック・ボックスが選択される場合、このフィールドはシミュレーション出力ファイルの最大数を指定します。デフォルトは **10** です。新規シミュレーション・ファイルが最大設定値を超える場合、最も古いシミュレーション・ファイルが削除されます。

「収集」ボタン
問題を診断するために IBM サポートが必要とするすべての成果物を収集するには、このボタンをクリックします。「問題判別情報の収集」ウィザードで、情報の収集対象となるアプリケーションのチェック・ボックスを選択し、宛先 .zip ファイルを指定します。以下の情報が収集され、宛先 .zip ファイルに集められます。

- 選択されたすべてのアプリケーションの内容
- ランタイム拡張プラグインの内容で、トレース、ログ、および runtime.properties ファイルを含む
- 現行のデータ・ディレクトリーからの .log ファイル

「デフォルトの復元」ボタン
パネル内のすべての設定をデフォルトに戻すには、このボタンをクリックします。

「適用」ボタン
ランタイム拡張プラグインの runtime.properties ファイルの設定を保存するには、このボタンをクリックします。

HATS リッチ・クライアントの考慮事項と制限

HATS リッチ・クライアント・アプリケーションを使用する際の考慮事項と制限のリストを次に示します。

- jclDesktop JRE は、Lotus Expeditor Toolkit バージョン 6.2.0 以前によってインストールされます。 この JRE には、HATS リッチ・クライアント・サポートが依存する Swing および AWT ライブラリーは含まれません。 HATS がワークスペースにこの JRE を検出すると、現行の JRE は HATS と非互換であることを示す警告ダイアログが表示され、JRE をデフォルトの Eclipse JRE に変更するオプションが提供されます。JRE を変更しない方を選択した場合、それを手動で変更する必要があります。**「JRE の非互換性に関する警告を表示」**設定を使用して、警告ダイアログを表示または抑止します。詳細については、 149 ページの『HATS プリファレンスの使用』を参照してください。
- リッチ・クライアント・アプリケーションで、テキストをイメージに置換するテキスト置換機能の使用と、特定の SWT ウィジェットの使用を組み合わせること

に関する制限があります。詳しくは、 195 ページの『テキスト置換』を参照してください。ウィジェットおよび制限は以下にリストされています。

- ボタン - 表題テキストのある部分がイメージに置換される場合、その他のすべての表題テキストは廃棄されます。
- ボタン・テーブル - ボタンと同じ。
- チェック・ボックス - ボタンと同じ。
- コンボ - ドロップダウン自体ではイメージの置換は行われません。イメージの置換は表題テキストについて行われます。イメージに置換されないその他のテキストは保存され、表題に表示されます。
- ドロップダウン (データ入力) およびドロップダウン (選択) - コンボと同じ。
- グラフ - イメージの置換は行われません。
- リンク - イメージの置換は行われません。
- リスト - コンボと同じ。
- ポップアップ - コンボと同じ。
- ラジオ・ボタン (データ入力) およびラジオ・ボタン (選択) - ボタンと同じ。
- サブファイル (チェック・ボックス) - イメージの置換はテーブル内で行われます。テーブルの上または下にあるボタンの表題テキストのある部分がイメージに置換される場合、その他のすべての表題テキストは廃棄されます。
- サブファイル (ドロップダウン) - イメージの置換はテーブル内でのみ行われます。イメージの置換はドロップダウン内では行われません。
- テキスト入力 - コンボと同じ。
- ツールバー - イメージは定義済みのイメージ・マッピングによって決定されるため、ツールバー・ウィジェットの使用時にはテキストとイメージの置換は行われません。

- Visual Editor を使用して SWT フォント、イメージ、および色のリソースを手動で HATS 変換およびテンプレートに追加する際に SWT リソース・リークを発生させる可能性がある制限があります。

  この制限があるため、フォント、イメージ、または色のリソースを HATS リッチ・クライアントのテンプレートまたは変換に追加する場合は、クリーンアップ・ロジックを手動で追加する必要があります。システムの色を追加する場合にクリーンアップ・ロジックを追加する必要はありません。これはシステムによって割り振られます。

  例えば、Visual Editor を使用して新規の色リソースを追加し、以下のコードが生成される場合、

  ```
  sideBarArea.setBackground(new Color(Display.getCurrent(), 221, 228, 255));
  ```

  以下のコード行を手動で追加して、ウィジェットが破棄されるときに色をクリーンアップする必要があります。

  ```
  sideBarArea.addDisposeListener(new DisposeListener(){
           public void widgetDisposed(DisposeEvent e){
               Color background = sideBarArea.getBackground();
               if (background != null && !background.isDisposed()) {
  ```

```
                background.dispose();
            }
        }
    });
```

HATS が提供する ColorManager クラスを使用する場合、色リソースをクリーンアップする必要はありません。しかし、Visual Editor が提供する Properties Editor の代わりに Java エディターを使用してコードを作成する必要があります。以下は、ColorManager を使用するサンプル・コードです。

```
sideBarArea.setBackground(com.ibm.hats.rcp.ui.misc.ColorManager.
      getInstance(Display.getCurrent()).getColor(221, 228, 255));
```

別の例として、Visual Editor を使用してイメージをテンプレートに手動で追加する場合、イメージを追加するコードとイメージ・リソースをクリーンアップするコードは以下のようになります。

```
label = new Label(bannerArea, SWT.RIGHT);
        gridData = new GridData(SWT.END, SWT.CENTER, true, true);
        label.setImage(new Image(Display.getCurrent(),
              getClass().getResourceAsStream("/images/myimage.gif")));
        label.setLayoutData(gridData);
        label.addDisposeListener(new DisposeListener() {
            public void widgetDisposed(DisposeEvent e){
                Image image = label.getImage();
                if (image != null && !image.isDisposed()) {
                    image.dispose();
                }
            }
        });
```

第 5 章 HATS プロジェクトの変更

HATS の「プロジェクトの作成」ウィザードを使用してプロジェクトを作成すると、ウィザードで選択した設定は、プロジェクト・アプリケーション・ファイル (.hap) に保存されます。プロジェクト・エディターを起動するには、「**HATS プロジェクト**」ビューで、変更するプロジェクトの名前の下の「**プロジェクト設定**」をダブルクリックします。

プロジェクト・エディターで表示される設定は、プロジェクト全体に使用される設定です。これらの設定のいずれかを変更する場合は、プロジェクト・エディターのタブを使用します。プロジェクト・エディターで変更した内容を保存すると、テスト環境でプロジェクトを実行する際に、アプリケーション・キーパッドで「**最新表示**」をクリックするか、 GUI で新しいホスト画面を表示することで、その内容が自動的に認識されます。「**最新表示**」では接続 (.hco) ファイルに対する変更内容が反映されません。接続設定に対する変更は、アプリケーション・サーバー (またはアプリケーション) が再始動すると、適用されます。

注: HATS プロジェクト全体に使用する設定を、HATS 全体に適用する設定と混同しないでください。HATS 全体の設定は、「**ウィンドウ**」>「**設定**」>「**HATS**」から適用できます。

以下のセクションでは、プロジェクト・エディターの各タブと、それらのタブを使用してプロジェクト設定を変更する方法について説明します。

概要

プロジェクト・エディターの「**概要**」タブには、プロジェクトを作成するときに指定したすべての設定の要約が示されます。

「**一般情報**」セクションには、プロジェクト名と説明、選択されているテンプレートとテーマ、および最終変更日時が記載されています。

注: モバイル装置用に最適化されたプロジェクトの場合、テーマは表示されません。詳しくは、53 ページの『モバイル装置用の HATS アプリケーションの作成』を参照してください。

プロジェクトのテーマを変更するには、現在のテーマのリンクをクリックします。提供されているオプションは、「**標準**」、「**最新の Web アプリケーション**」 Web のみ または「**最新の UI アプリケーション**」 RCP のみ、「**従来型端末エミュレーター**」、および「**カスタム**」の 4 つです。設定リストの各機能は、ほかのプロジェクト設定でも有効にできます。このリストは、アプリケーションの外観や振る舞いをテーマとしてまとめて扱うことのできる機能を 1 か所で設定する場合に便利です。

次の表は、各テーマ設定がそれぞれどのようにプロジェクト設定にマップされるかを示したものです。

© Copyright IBM Corp. 2003, 2015

表 1. テーマ設定のプロジェクト設定へのマップ

テーマ設定	プロジェクト・エディター・タブ	プロジェクト設定
矢印キー・ナビゲーション RCP のみ	その他	「クライアント設定」>「矢印キー・ナビゲーションを使用可能にする」
自動フィールド進行	その他	「クライアント設定」>「自動フィールド進行を使用可能にする」
保護フィールド上にあるカーソル	レンダリング	「ウィジェット」>「フィールド」>「保護フィールド上でのカーソル位置決めを許可します」フィールド
フィールド拡張属性	レンダリング	「ウィジェット」>「フィールド」>「拡張属性を使用可能にする」
ホスト・キーパッド	レンダリング	「ホスト・キーパッド」>「デフォルトのホスト・キーパッドを表示」
キーボード・サポート	その他	「キーボード・サポート」>「キーボード・サポートを使用可能にする」
オペレーター情報域	レンダリング	「オペレーター情報域」>「OIA を表示」
上書きモード (初期)	その他	「クライアント設定」>「上書きモード (初期)」
モノスペース・フォントを使用したレンダリング	レンダリング	「ウィジェット」>「フィールド」>「モノスペース・フォントを使用してレンダリング」

「標準」のテーマで使用できるのは、HATS プロジェクトを最初に作成したときにデフォルトで有効になっている機能です。「最新」のテーマを選択すると、個別の機能はすべて選択解除されます。「従来型」のテーマを選択すると、すべての機能が選択されます。「カスタム」のテーマを選択すると、機能の組み合わせを任意に選択できます。

注: DBCS サポートを使用する場合のテーマ設定について詳しくは、 544 ページの『プロジェクト・テーマ設定』を参照してください。

「**テスト**」セクションには、プロジェクトのテストに使用できるリンクが含まれています。HATS Web プロジェクトの場合は、次のようなリンクがあります。

- WebSphere Application Server 上で起動
- WebSphere Application Server 上でデバッグ・モードで起動

HATS リッチ・クライアント・プロジェクトの場合は、次のようなリンクがあります。

- ターゲット・プラットフォーム上で起動
- ターゲット・プラットフォーム上でデバッグ・モードで起動

注:

1. これらのリンクのいずれかをクリックすると、「HATS Projects」ビューでプロジェクトを右クリックし、「**サーバーで実行**」 (リッチ・クライアントの場合は「**実行**」) または「**サーバーでデバッグ**」 (リッチ・クライアントの場合は「**デバッグ**」) を選択した場合と同じ操作が実行されます。

2. モードのテストについて詳しくは、*「HATS スタートアップ・ガイド」* の『プロジェクトのテスト』を参照してください。
3. HATS リッチ・クライアント・アプリケーションのテストについて詳しくは、74 ページの『HATS リッチ・クライアント・アプリケーションのテスト』を参照してください。

「デプロイメント」 RCP のみ セクションには、**「ターゲット・プラットフォーム」**ドロップダウンがあります。リッチ・クライアント・プロジェクトのターゲット・デプロイメントを変更するには、このドロップダウンを使用します。これにより、あるプラットフォームをターゲットとしてビルドを開始したアプリケーションを、後で別のプラットフォーム用に切り替えることができます。サポートされるプラットフォームは、**Eclipse リッチ・クライアント・プラットフォーム**、**Lotus Expeditor Client for Desktop**、および **Lotus Notes** です。

ここで、ターゲット・プラットフォームを変更し、それがワークスペースのターゲット・プラットフォームに一致しない場合、これを通知するメッセージが表示されます。この場合に実行するアクションは、このプロジェクトについて変更したプラットフォームによって異なります。

Lotus Notes または Lotus Expeditor Client プラットフォームに変更した場合は、74 ページの『ターゲット・プラットフォームの構成』を参照し、ターゲット・プラットフォームの設定に関する情報を確認してください。

Eclipse リッチ・クライアント・プラットフォームに変更した場合は、**「ウィンドウ」>「設定」>「プラグイン開発 (Plugin-in Development)」>「ターゲット・プラットフォーム」**を選択し、**「場所」**ドロップダウン・リストから Rational SDP インストール・ディレクトリー(デフォルトは C:¥Program Files¥IBM¥SDP) を選択します。

ターゲット・プラットフォームを再ロードするようにプロンプトが出されたら、**「はい」**を選択します。

また、**「デプロイメント」**セクションには**「プラグイン・マニフェストをオープンしてください」**というリンクもあり、これで、プロジェクトのプラグイン記述子 (plugin.xml) のエディターが起動します。拡張リッチ・クライアントを開発する場合は、このファイルを変更する必要があります。

「接続」セクション、**「レンダリング設定」**セクション、および**「各種設定」**セクションは、それぞれの対応するタブで指定されている設定を要約します。見出し自体が、以下のセクションで説明する対応するタブへのリンクです。

接続

「接続」タブには、プロジェクト接続に関する情報が表示されます。HATS には、デフォルト接続 (変換接続とも呼ばれる) とバックグラウンド接続の 2 つの接続のタイプがあります。各 HATS アプリケーションでは、HATS によって変換される画面を持つホスト・アプリケーション用のデフォルト接続を 1 つ保持しています。バックグラウンド接続とは、プロジェクト内にある、デフォルト接続以外のすべての接続です。 HATS では、バックグラウンド接続からの画面の変換は実行されません。詳しくは、 157 ページの『第 6 章 接続の管理』を参照してください。

「**デフォルト**」ドロップダウン・リストからデフォルト・ホスト接続を選択できます。

また、すべてのプロジェクト接続を表示したリスト・テーブルもあります。リストには、接続名、ホスト、ホスト・タイプ、ポート番号、およびコード・ページが表示されます。「**追加**」ボタンをクリックして追加のプロジェクト接続を追加すると、「**接続の作成**」ウィザードが起動します。HATS では、プロジェクトに使用可能な接続を複数指定できます。ウィザードの使用の詳細については、157 ページの『接続の作成』を参照してください。

注: 双方向 (bidi) コード・ページを選択した場合は、538 ページの『HATS ファイルへの追加』を参照してください。

既存の接続を編集するには、接続を選択して「**編集**」をクリックし、接続エディターを起動する必要があります。詳しくは、158 ページの『接続エディター』を参照してください。

「**除去**」ボタンをクリックすると、強調表示された接続が削除されます。「**最新表示**」をクリックすると、リストが更新されます。

テンプレート

「**テンプレート**」タブには、変換を囲むためのテンプレートが表示されます。

このタブでは、デフォルト・テンプレートとしていずれのテンプレートを使用するかを変更することができます。画面カスタマイズの操作を使用して特定の画面に異なるテンプレートを指定しない限り、これがアプリケーションのすべての画面に使用されます。

注: 新規に提供されたテンプレートには、アスタリスクでマークが付けられています。

デフォルト・テンプレートは、プロジェクトのすべての変換に適用されるテンプレートとなり、不一致画面イベントのデフォルトのアクションとしてデフォルトの変換を使用してこのテンプレートが適用されます。不一致画面イベントのアクションを変更する方法については、125 ページの『アプリケーション・イベント』を参照してください。

画面カスタマイズのアクションを作成または変更するときに、別のテンプレートを選択してデフォルト・テンプレートをオーバーライドすることができます。「**テンプレート**」タブでテンプレートを編集することもできます。

HATS プロジェクト用のテンプレートを独自に作成できます。詳しくは、371 ページの『「テンプレートの作成」ウィザード』を参照してください。

レンダリング

「**レンダリング**」タブには、デフォルト・レンダリング、グローバル規則、テキスト置換、コンポーネント、ウィジェット、ツールバー RCP のみ、アプリケーション・キーパッド、ホスト・キーパッド、およびオペレーター情報域の設定が表示されます。ここでは、HATS プロジェクトのプロジェクト・レベルのデフォルト値を構成できます。

デフォルト・レンダリング

デフォルト・レンダリングは、個別に変換されなかったホスト画面を変換するためのものです。選択済みのデフォルト・レンダリング・セットは、元のホスト画面の構造を保持する一方、GUI の設計の原則を適用することによってアプリケーションの機能を拡張しようとします。つまり、ホスト画面の無保護フィールドを GUI の入力フィールドとしてレンダリングするだけでなく、ファンクション・キーをボタンやリンクに変換したり、選択リストをドロップダウンに変換したり、テーブル領域をテーブルに変換したりすることもできます。

注:

1. ホスト画面用の個々の変換を作成する場合は、デフォルト・レンダリングを事前にその変換に取り込むことができます。詳しくは、203 ページの『「変換の作成」ウィザード』を参照してください。
2. また、デフォルト・レンダリングをあとで変換に挿入することもできます。詳しくは、215 ページの『デフォルト・レンダリングを挿入』を参照してください。

レンダリング・セットは、レンダリング項目の優先順位付きリストを作成することによって構成します。各レンダリング項目は、指定のホスト・コンポーネントを認識し、指定のウィジェットを使用してレンダリングする特定の領域を定義します。例えば、ホスト画面の下部領域にあるファンクション・キーを探し、リンクとしてレンダリングできます。

デフォルトの変換に対して複数のレンダリング・セットを作成できます。レンダリング・セットを追加で作成した場合でも、デフォルトの変換で使用されるレンダリング・セットは、1 つだけです。新規セットを作成するには、「**追加**」ボタンをクリックして、「**レンダリング・セットを追加**」ウィンドウを開きます。新しいレンダリング・セットに対して固有の「**名前**」と 「**説明 (オプション)**」を指定します。

新規レンダリング・セットをデフォルトにするには、「**このレンダリング・セットをデフォルト・レンダリングに使用**」チェック・ボックスを選択します。

「**空のレンダリング・セットを作成**」ボタンを選択してレンダリング・セットをブランクにする (レンダリング項目を定義しない) ことも、「**新規レンダリング・セットを既存のセットからコピー**」プルダウン・メニューからセットを選択することもできます。リストの横にある「**編集**」を選択してレンダリング・セットを変更することも、「**除去**」ボタンを選択して削除することもできます。

以下のリストにあるレンダリング項目は、デフォルト・レンダリング・セットの一部として定義されます。

- ダイアログ
- ENPTUI ウィンドウ

- サブファイル
- ENPTUI メニュー・バー
- ENPTUI アクション
- ENPTUI 単一選択
- ENPTUI 複数選択
- ENPTUI スクロール・バー
- ライト・ペン - アテンション
- ライト・ペン - 選択
- 選択リスト
- ファンクション・キー
- フィールド・テーブル
- 表示されるテーブル
- URL
- 残りのテキストおよび入力フィールド

注:

1. ENPTUI レンダリング項目は、使用するホストが 5250 または 5250W で、HATS プロジェクトの初期作成時に「**グラフィカル・インターフェース DDS キーワード (ENPTUI) レンダリング・サポートの追加**」チェック・ボックスを選択して ENPTUI サポートを使用可能にした場合にのみ表示されます。
2. ライト・ペン・レンダリング項目は、使用するホストが 3270 または 3270E で、 HATS プロジェクトの初期作成時に「**ライト・ペン・レンダリング・サポートの追加**」チェック・ボックスを選択してライト・ペン・サポートを使用可能にした場合にのみ表示されます。

このプロジェクトのすべての画面について、選択済みレンダリング・セットに新規レンダリング項目を追加できます。また、編集または除去する既存のレンダリング項目を選択し、テーブルの右側にある該当するボタンを選択して、これらの既存のレンダリング項目を編集または除去することもできます。

「**追加**」ボタンまたは「**編集**」ボタンを選択した場合は、レンダリング項目の名前および説明を入力します。次に、レンダリング項目を適用する画面領域を定義し、認識に使用するホスト・コンポーネントおよびレンダリングに使用するウィジェットを指定します。また、レンダリング項目の適用順序を変更することもできます。それには、リストで項目を選択し、「**上へ**」ボタンまたは「**下へ**」ボタンをクリックしてリストの上または下に移動させます。ウィザードの最後の 2 つのパネルについては、213 ページの『ホスト・コンポーネントを挿入』または 215 ページの『ホスト・コンポーネントを編集Web のみ』を参照してください。

注:

1. Dojo ウィジェットは、デフォルト・レンダリング項目ではサポートされていません。
2. ダイアログ・コンポーネントは、ホスト画面でモーダル・フレーム (ポップアップ・フレーム) を認識してレンダリングするための特殊なコンポーネントです。ただし、ダイアログ・コンポーネントを使用して、モーダル・フレーム・ホスト・コンポーネントを各変換に挿入することはできません。

レンダリング・セットを構成するレンダリング項目のリストは番号付きリストです。 リストの下にある別の項目によっても認識されるホスト画面の一部を各レンダリング項目が消費する場合があるため、リストの順序が重要です。レンダリング項目は、リストの上にあるほど優先順位が高くなります。各レンダリング項目の隣にあるチェック・ボックスを使用すると、選択を使用可能または使用不可にすることができます。

コンポーネントは、必ずしも指定した領域全体を消費するとは限りません。例えば、画面全体で選択リストを検索するように選択リスト・コンポーネントを構成したが、画面の中央で 1 つのリストしか検出されなかったとします。ホスト画面の中央の領域のみが「消費済み」としてマークされます。ホスト画面の残りの「未消費」領域は、引き続き優先順位の低いレンダリング項目によって変換されます。

デフォルト・レンダリング設定でテーブル・コンポーネントを使用する場合は注意してください。これは、テーブル・コンポーネントは選択されている領域でほぼすべての画面を認識し、リスト内のこれより下位のレンダリング項目が認識されないためです。

デフォルト・リストの最後のレンダリング項目 (**「残りのテキストおよび入力フィールド」**) は、残りのすべての「未消費」画面領域を、フィールド・コンポーネントおよびフィールド・ウィジェットを使用して変換します。この項目は、レンダリング・セットの最後の項目でなければなりません。

通常、デフォルト・レンダリング・セットは、GUI の設計原則を適用することによってアプリケーションの機能を拡張しながら、元のホスト画面の構造を保持しようとします。しかし、ホスト画面のデフォルト・レンダリングをモバイル装置上に表示するために、ある程度のコンパクト化が必要になる場合があります。コンパクト化の過程で HTML およびブランク・スペースの量が減らされた結果、元のホスト画面とは異なる構造の表示になる可能性があります。現在選択されているレンダリング・セットでコンパクト化を使用するように指定するには、**「コンパクト・レンダリングの使用」** **Web のみ** を選択します。

注: モバイル装置用に最適化された Web プロジェクトには、この設定が選択された場合のデフォルトとして、**「コンパクト (compact)」**というレンダリング・セットが作成されます。

複数のレンダリング・セットの例

HATS アプリケーションを 1 つ作成し、ユーザーがそのアプリケーションから、互いにまったく異なるように見える 2 つのホスト・アプリケーションにアクセスできるようにしたいと仮定します。この時、各アプリケーションに異なるレンダリング・セットを使用する一方で、HATS のデフォルトのレンダリングを活用して固有の画面カスタマイズをできるだけ少なくしたいと考えたとします。

その場合は、以下の操作によって、これら 2 つのアプリケーション APPA と APPB を呼び出します。

1. APPA 用に調整したレンダリング・セットを作成します。
2. APPB 用に調整したレンダリング・セットを作成します。
3. APPA レンダリング・セットのパラメーターを使用したデフォルトのレンダリング・タグを持つ変換「APPA」を作成します。

4. APPB レンダリング・セットのパラメーターを使用したデフォルトのレンダリング・タグを持つ変換「APPB」を作成します。
5. APPA の最初の画面だけを認識し、以下のアクションを実行する画面カスタマイズ APPAfirst を作成します。
   a. グローバル変数「WhichApp」を「APPA」に設定します。
   b. 変換 APPA を適用します。
6. APPB の最初の画面だけを認識し、以下のアクションを実行する画面カスタマイズ APPBfirst を作成します。
   a. グローバル変数「WhichApp」を「APPB」に設定します。
   b. 変換 APPB を適用します。
7. グローバル変数「WhichApp」が「APPA」である場合に真であることを認識し、アクションとして変換 APPA を適用する、画面カスタマイズ APPArest を作成します。
8. グローバル変数「WhichApp」が「APPB」である場合に真であることを認識し、アクションとして変換 APPB を適用する、画面カスタマイズ APPBrest を作成します。
9. プロジェクト設定の「**イベント**」タブで画面カスタマイズを APPAfirst、APPBfirst、APPArest、APPBrest の順に並べます。

拡張レンダリング

拡張レンダリング設定を変更するには、デフォルト・レンダリング・ツリーを展開して「**拡張**」をクリックします。

代替レンダリング: このセクションでは、HATS コンポーネントの変換時にレンダリングの対象が認識されない場合、デフォルト・レンダリングを使用するように指定することができます。この機能は、アプリケーションに含まれる画面に、無保護状態と保護状態を切り替えられるフィールドがある場合には特に便利です。例えば、この機能を使用せず、入力フィールド・コンポーネントによって領域を認識する場合は、フィールドが保護属性状態に切り替わると何も認識されず、レンダリングもされません。

コンポーネント・レンダリングを失敗した際にデフォルト・レンダリングを使用

ホスト・コンポーネントが画面イベント内の領域を認識しない場合に実行されるアクションをプロジェクト・レベルで指定するには、このオプションを使用します。このオプションはデフォルトでは選択されていません。その結果、指定したコンポーネントが領域を認識しない場合は何もレンダリングされません。このオプションを選択すると、領域はデフォルト・レンダリングを使用して変換されます。

レンダリング・セット

このドロップダウンでは、現在定義されているレンダリング・セットのリストから、変換で使用するレンダリング・セットを選択します。

注:

1. 選択したレンダリング・セットが後でプロジェクトから削除され、実行時には存在しなくなっている場合は、デフォルトのレンダリング・セットが使用されます。

2. この機能は、画面変換の個々のコンポーネント・レベルで指定することもできます。詳しくは、213 ページの『ホスト・コンポーネントを挿入』を参照してください。
3. Dojo ウィジェットは、代替レンダリング・セットではサポートされていません。

HTML テーブル: このセクションでは、整形式の HTML をデフォルト・レンダリングで作成するオプションを選択できます。

デフォルト・レンダリングのテーブル・データと行のタグのクローズ

デフォルト・レンダリングにあるすべてのテーブル・データおよびテーブル行のタグを閉じる (つまり、</td> および </tr> を付ける) ようにプロジェクト・レベルで指定するには、このオプションを選択します。

デフォルト・レンダリングに含まれる HTML の量を減らすために、デフォルトではこのオプションは選択されていません。パフォーマンスが重要になるアプリケーションでは、HTML の量を減らすことが大きな意味を持ちます。しかし、アクセシビリティー用に使用されるスクリーン・リーダー (読み上げソフトウェア) アプリケーションなど、他のアプリケーションではこのオプションの選択が必要になる場合があります。

新規 HATS プロジェクトでデフォルトのレンダリング・カスタマイズを使用可能にする方法

次回に新規 HATS プロジェクトを作成するときにデフォルトのレンダリング・カスタマイズを表示するには、次のステップを実行します。

1. プロジェクトを作成し、デフォルトのレンダリング・オプションをカスタマイズします。
2. このプロジェクトの application.hap ファイルを開きます。Web プロジェクトの場合、このファイルは ¥*workspace_directory*¥project name¥Web Content¥WEB-INF¥profiles¥ にあります。リッチ・クライアント・プロジェクトの場合、このファイルは ¥*workspace_directory*¥project name¥profiles¥ にあります。
3. application.hap ファイルの <defaultRendering> タグから </defaultRendering> タグまでのソースをコピーし、インストールされた HATS オファリングの下にある application.hap ファイルの <defaultRendering> タグから </defaultRendering> タグまでのソースに貼り付けます。

   Web プロジェクトの場合、完全なファイル・パスは次のとおりです。

   ```
   <shared_install_directory>¥plugins¥com.ibm.hats_nnn¥predefined¥
   projects¥new¥Web Content¥WEB-INF¥profiles¥application.hap
   ```

   リッチ・クライアント・プロジェクトの場合、完全なファイル・パスは次のとおりです。

   ```
   <shared_install_directory>¥plugins¥com.ibm.hats_nnn¥predefined¥
   projects¥rcp¥new¥profiles¥application.hap
   ```

   上記のファイル・パスで、*shared_install_directory* は IBM Installation Manager を使用して HATS オファリングをインストールした共用リソース・ディレクトリー、*nnn* は HATS のバージョンおよびビルド・レベルです。

注: HATS 保守のインストール後に、この手順を繰り返す必要があります。新規の application.hap ファイルは新しい com.ibm.hats_nnn ディレクトリー構造の下に置かれます。

グローバル規則

グローバル規則を使用すると、ホスト入力フィールドのパターン認識と変換が可能になります。また、グローバル規則では、カスタマイズした画面、およびカスタマイズしていない (デフォルト・レンダリングの) 画面が処理されます。グローバル規則は、プロジェクト・レベルと、画面レベルの両方で定義できます。プロジェクト・レベルのグローバル規則を指定する場合は、このセクションを使用します。画面レベルのグローバル規則を定義する場合については、194 ページの『グローバル規則』を参照してください。

同じ入力フィールドに対してプロジェクト・レベルのグローバル規則と画面レベルのグローバル規則の両方が定義されている場合は、画面レベルの規則が優先されます。

例えば、「Country」というワードが前にあるすべての入力フィールドを、世界各国のコードを含むドロップダウンとしてレンダリングするグローバル規則を作成できます。プロジェクト・レベルのグローバル規則は HATS アプリケーションに対してグローバルです。これらの規則は、レンダリング設定で変換された画面とカスタム変換で変換された画面の両方で機能します。これにより、アプリケーション全体で、変換を一切変更せずにホスト画面の特定のパターンを認識させることができます。

グローバル規則は、構成されたパターン・タイプおよび変換フラグメントから構成されます。パターン・タイプ構成は、ホスト画面で検索するコンテンツの種類を指定します。この操作によって、特定のサイズの特定のフィールドまたはすべてのフィールドを変換したり、指定したストリングに最も近いフィールドを変換できます。例えば、前に「Date」という語がある入力フィールドを認識対象パターンにすることができます。変換フラグメントには、すべての変換におけるパターンのすべての出現箇所を置換する場合に使用されるコンテンツが含まれています。変換フラグメントには、HATS 入力フィールド・コンポーネントおよびカレンダー・ウィジェットを入れることができます。これにより、アプリケーション全体で、前に「Date」という語があるすべての入力フィールドをカレンダー・コントロールとして表示できます。

構成されているグローバル規則は、すべてグローバル規則のテーブルに表示されます。グローバル規則は、すべての「**変換を適用**」アクションの先頭からこのリストに表示される順序で処理されます。入力フィールドがリストの 1 つのグローバル規則によって認識されると、以降のグローバル規則では認識されなくなります。各項目の隣にあるチェック・ボックスには、項目が使用可能か使用不可かが示されます。

1. このプロジェクトのグローバル規則のリストにグローバル規則定義を追加するには、「**追加**」ボタンをクリックします。画面キャプチャーがない場合は、ホスト端末をオープンして画面を取り込むように要求されます。新しいグローバル規則を作成するには、「**名前**」、「**パッケージ**」 (リッチ・クライアント・プロジェクトの場合)、および「**説明**」を入力し、このグローバル規則を「**新規変換フラ**

グメントを作成」と関連付けます。グローバル規則を編集する場合は、「**既存の変換フラグメントの使用 (Use an existing transformation fragment)**」を実行できます。

2. 次に、使用するパターン・タイプを選択します。選択できるパターン・タイプは以下のとおりです。
   - 「**周囲のテキストによって入力フィールドを検出**」のパターン・タイプは、指定のストリングが左、右、上、または下の境界となっているすべての入力フィールドを認識します。このパターン・タイプでは、個別の「**パターン設定 (Pattern Settings)** 」を選択する必要があります。指定するストリングには、ワイルドカードとして 1 つ以上のアスタリスク (*) を使用できます。スペースも有効です。最も近い入力フィールドのみまたはすべての入力フィールドを「**変換**」できます。グローバル規則を適用するために、テキスト・パターンが入力フィールドに対してどの位置になければいけないかを「**位置**」ドロップダウン・リストで、指定します。「**保護フィールドの内容**」フィールドに、このグローバル規則を適用するテキストを挿入します。「**大文字と小文字を区別**」チェック・ボックスを使用すると、指定したテキストが厳密に検索されます。このパターン・タイプは、上記の「Country」および「Date」の例で使用しています。

     「**変換**」オプションで「**すべての入力フィールド**」を選択すると、基準に一致する最初の入力フィールドによって、それ以後のフィールドの検索領域の有効範囲が決まります。次の例では、Enter. というストリングを含む保護フィールドの下にあるすべての入力フィールドに一致すると見なされるフィールドが表示されています。「グローバル規則設定」ウィザードでは、一致するフィールドがすべて強調表示されます。最初に一致したフィールドには、Alpha field (blue) というキャプションが付きます。このフィールドの開始列と終了列によって、その下にあるフィールドのうち、同様に一致と見なされるフィールドが決まります。キャプションが Numeric field (green) のフィールドと Column separators のフィールドは、最初に一致したフィールドより長かったり短かったりするので、一致とは見なされません。キャプションが Date field (pink) のフィールドと、そのすぐ下の 3 つのフィールドは、開始列と終了列が最初に一致したフィールドと同じなので、一致と見なされます。

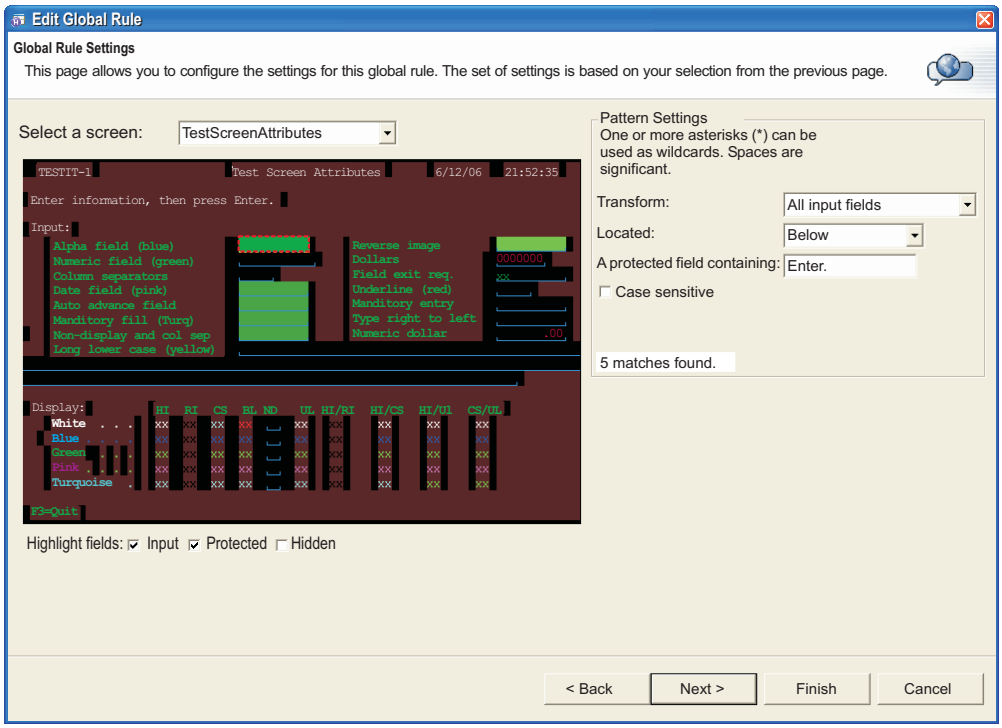


図 11. HATS グローバル規則の例

- 「**すべての入力フィールドを検出**」のパターン・タイプは、ホスト画面のすべての入力フィールドを正常に認識します。このパターン・タイプにパターン設定はありません。このパターン・タイプを使用するグローバル規則は、グローバル規則リストの最後に置き、優先順位を最も低くする必要があります。
- 「**フィールド・サイズによって入力フィールドを検出**」のパターン・タイプは、指定の長さのすべての入力フィールドを認識します。例えば、長さ 1 の入力フィールドをチェック・ボックス・コントロールに変換する場合に有用です。
- 「**指定した領域内で入力フィールドを検索**」のパターン・タイプは、ホスト画面領域内にあり、開始行、開始列、終了行、終了列で指定した任意の入力フィールドを認識します。オプションで、フィールド長を、その領域内の入力フィールドを詳細に定義するように指定することもできます。ホスト・フィールドは、認識される指定領域内に完全に含まれていなければなりません。このパターン・タイプを画面レベルのグローバル規則とあわせて使用すると、画面上でカスタマイズする入力フィールド (例えば、テーブル内のデータ) を特定する一方で、その画面のデフォルト・レンダリングを使用することもできます。

各パターン・タイプ・コンポーネントは、上記の一連のカスタマイズ可能な設定を含むことができます。設定を変更すると、左側の画面が更新され、一致する入力フィールドが反映されます (複数のフィールドが一致した場合は、複数のフィールドが強調表示されます)。「**強調表示するフィールド**」オプション以外を選択して、ホスト画面で特定のフィールドを強調表示することもできます。入力フィールドが画面上で定義されている位置を表示するには、「**入力**」チェック・ボックスを選択します。保護フィールドを表示する場合は、「**保護**」チェック・ボックスを選択します。隠しフィールドを強調表示したい場合は、「**非表示**」チェック・ボックスを選択します。入力フィールド、保護フィールド、または隠しフ

ィールドの強調表示色を変更する方法については、 149 ページの『HATS プリファレンスの使用』を参照してください。既存のフラグメントを使用している場合は、「**完了**」をクリックしてグローバル規則セットを更新し、下記のセクションをスキップします。

3. 新規フラグメントを作成している場合は、ここで変換フラグメントに挿入する HATS コンポーネントを構成する必要があります。グローバル規則の目的は入力フィールドの検出であるため、入力コンポーネントのみが表示されます。このページの構成に使用するホスト画面領域は、直前のページで最初に一致した入力フィールドを囲む選択ボックスによって示されますが、プロジェクト・レベルのグローバル規則は、このホスト画面とその他すべての画面で一致したすべての入力フィールドに対して影響します。コンポーネントおよびウィジェットを選択して、設定を構成します。詳細については、 213 ページの『ホスト・コンポーネントを挿入』または 215 ページの『ホスト・コンポーネントを編集Web のみ』を参照してください。 「**完了**」をクリックすると、グローバル規則セットが更新され、新規変換フラグメントが「**変換フラグメント**」フォルダーに作成されます。ウィザードを完了したら、必ずプロジェクト設定を保存してください。

注:

1. Dojo ウィジェットは、グローバル規則変換フラグメントではサポートされていません。また、グローバル規則は Dojo ウィジェット内では使用されません。
2. リッチ・クライアント・プロジェクトでは、SWT テーブル・ウィジェット付きの入力フィールド・コンポーネントは、グローバル規則ではレンダリングできません。

既存のグローバル規則のコンポーネント設定およびウィジェット設定を変更する必要がある場合は、「**レンダリング**」タブで「**グローバル規則**」を選択し、リストから変換フラグメントを選択して、「**編集**」をクリックします。

最初、変換フラグメントは、「グローバル規則」ウィザードの 3 ページ目で選択した設定に対応する特殊な HATS コンポーネントで作成されます。このコンポーネントは、領域属性を持たない点で特殊です。代わりに、レンダリングする領域を実行時に動的に取得します。このコンポーネントはほかの HATS コンポーネントと同様に編集できますが、ウィザードの最初のページで画面領域ではなくグローバル規則を選択する必要があります。この変換フラグメントには、他の有効な HATS 変換コンテンツを挿入することもできます。

注: グローバル規則を作成して変換フラグメントを編集する場合は、「設計」タブと「プレビュー」タブのどちらでも出力を表示できません。出力を表示するには、変換フラグメントまたはグローバル規則を適用する全変換の編集中に、「ホスト・コンポーネントを編集」機能から全ページのプレビューを使用します。

グローバル規則に関連付けられた変換フラグメントを手動で変換する場合は、「**レンダリング**」タブからエディターを使用する前に手動変換してください。こうすると、誤ったタグが読み取られ、すべての変更が削除されます。

プロジェクト・レベルのグローバル規則と該当する画面レベルのグローバル規則は、どちらも画面イベント変換で適用されます。画面イベント・レベルでは、グローバル規則処理をすべて使用不可にすることができます。使用不可にするには、次のステップを実行します。

1. 「**HATS Projects**」ビューで、画面イベントが入っている「**画面カスタマイズ**」フォルダーまたは「**画面組み合わせ**」フォルダーを開きます。
2. 画面イベントをダブルクリックします。
3. 「**アクション**」タブに進みます。
4. 「**変換を適用**」アクションを選択します。
5. 「**編集**」をクリックします。
6. 「**プロジェクト・レベルおよび画面レベルのグローバル規則を適用**」チェック・ボックスをクリアします。

テキスト置換

HATS アプリケーションでは、ホスト画面のテキスト・ストリングを別のテキスト・ストリング、HTML コンテンツ Web のみ、またはイメージに変換できます。

「**テキスト置換**」には、置換されるオリジナル・テキストとともに、置換に使用するテキスト、 HTML コンテンツ Web のみ、またはイメージを示すテーブルが表示されます。さらに、テキスト検索で大文字と小文字を区別するかどうか、正規表現サポートを使用するかどうかも表示されます。値のテーブルの右側にあるボタンを使用して、テキスト置換の仕様を追加、変更、または除去できます。これらの設定の指定方法の詳細については、 195 ページの『テキスト置換』を参照してください。

コンポーネントおよびウィジェット

個々のコンポーネントおよびウィジェットについて設定を変更するには、ツリー内で該当項目を展開して、個々のコンポーネントおよびウィジェットを表示します。コンポーネントおよびウィジェットには、カスタマイズ可能な設定がないものもあります。「ホスト・コンポーネントを挿入」ウィザードを使用してカスタマイズできる設定については、 223 ページの『コンポーネントおよびウィジェットの設定』を参照してください。

コンポーネントおよびウィジェットのプロジェクト設定を変換に挿入するときに、デフォルトのプロジェクト設定をオーバーライドすることができます。これらの変更された設定は、変換内の該当のコンポーネントまたはウィジェットの個別のインスタンスのみに適用されます。プロジェクト内のすべての変換について、そのコンポーネントまたはウィジェット以外のすべてのインスタンスでは、ユーザーが特に変更しない限り、デフォルトのプロジェクト設定が使用されます。例えば、VisualTable コンポーネントに関するデフォルト設定があります。1 つの変換に 2 つの VisualTable コンポーネントを配置し、片方のコンポーネントではプロジェクト設定のデフォルト設定を使用し、もう一方では変更した設定を使用することができます。

コンポーネント

「**コンポーネント**」には、ホスト画面のエレメントを GUI に表示可能なオブジェクトに変換するために使用するホスト・コンポーネントのリストが表示されます。ホ

スト・コンポーネントの設定では、そのコンポーネントがホスト画面で認識される方法を指定します。一部のコンポーネントは変更できます。ホスト・コンポーネントの詳細については、223 ページの『ホスト・コンポーネントの設定』を参照してください。

ウィジェット

「**ウィジェット**」を選択すると、ホスト・コンポーネントを GUI に表示するためにプロジェクトで使用できるすべての HATS ウィジェットのリストが表示されます。一部のウィジェットは変更できます。ウィジェットをカスタマイズするのは、ウィジェットが Web ページ上で表示される方法を指定するためです。変更可能なウィジェットおよび設定については、258 ページの『ウィジェット設定』を参照してください。

ツールバー RCP のみ

「ツールバー」設定パネルでは、リッチ・クライアント・アプリケーションの「変換」ビューのツールバーに関連した設定を変更できます。ツールバーには、アプリケーション・キーパッド・アクション (「最新表示」、「デフォルト」など) に加えて、ツールバー・ウィジェットが提供するアクションも配置されています。ツールバー・ウィジェットの詳細については、347 ページの『ツールバー RCP のみ』を参照してください。

HATS Web アプリケーションでは、アプリケーション・キーパッドは通常はテンプレートの下部またはサイド・バーに配置されます。HATS リッチ・クライアント・アプリケーションでは、アプリケーション・キーパッドにより提供される機能は「変換」ビューのツールバーにボタンとして配置されます。これによって、HATS リッチ・クライアント・アプリケーションを、リッチ・クライアント環境内の他のアプリケーションと適合させることができます。「変換」ビューのツールバーについて詳しくは、97 ページの『ツールバー』を参照してください。

このパネルでは、「**ツールバーを表示**」チェック・ボックスを選択したり、クリアしたりできます。選択する場合は、ツールバー項目の表示方法 (「**テキスト**」、「**イメージ**」、または「**テキストとイメージの両方**」) を設定できます。この設定は、ツールバーのすべての項目 (ツールバー・ウィジェットが提供する項目も含む) に適用されます。

アプリケーション・キーパッド

デフォルトのアプリケーション・キーパッドは、デフォルト・テンプレートに表示されます。アプリケーション・キーパッドについては、以下の設定をカスタマイズできます。

デフォルトのアプリケーション・キーパッドを表示

ユーザーがアプリケーションと対話するときに、テンプレートまたは変換に定義されたデフォルトのアプリケーション・キーパッドを GUI に表示させる場合は、このチェック・ボックスを選択します。

表示するキーの選択

「**デフォルトのアプリケーション・キーパッドを表示**」チェック・ボックスを選択した場合、GUI でデフォルトのアプリケーション・キーパッドに含める各キーの隣にあるチェック・ボックスを選択できます。

表示 **Web のみ**

選択したキーを、ボタン、リンク、アイコン、またはドロップダウン・リスト項目のいずれとして表示するかを決定する値を選択します。

ホスト・キーパッド

ホスト・キーパッドについては、以下の設定をカスタマイズできます。

デフォルトのホスト・キーパッドを表示

ユーザーがアプリケーションと対話するときに、テンプレートまたは変換に定義されたデフォルトのホスト・キーパッドを GUI に表示させる場合は、このチェック・ボックスを選択します。

表示するキーの選択

「**デフォルトのホスト・キーパッドを表示**」チェック・ボックスを選択した場合、GUI でデフォルトのホスト・キーパッドに含める各キーの隣にあるチェック・ボックスを選択できます。使用可能なすべてのキーを含めるには、「**すべて選択**」を選択します。選択されているすべてのボックスをクリアするには、「**全選択解除**」を選択します。

表示 選択したキーを、ボタン、リンク、またはドロップダウン・リスト項目のいずれとして表示するかを決定する値を、このドロップダウン・リストから選択します。

「**追加**」ボタンをクリックして「**表題**」および「**ニーモニック**」を入力することによって、カスタム・ホスト・キーを追加することもできます。作業が完了したら、「**OK**」をクリックします。

オペレーター情報域

オペレーター情報域 (OIA) を使用すると、ホスト・アプリケーションの状態を簡単に判断できます。OIA については、以下の設定をカスタマイズできます。

OIA の表示

このボックスを選択すると、OIA が表示されます。以下のチェック・ボックスを選択すると、表示する項目を指定できます。

セキュア・ホスト接続

HATS 接続を SSL で保護するかどうかを表示するには、このボックスを選択します。

入力禁止標識

キーボードがロックされており、キーボードからの入力を行えないかどうかを表示する場合は、このボックスを選択します。

システム・ロック標識

戻りデータを待機する間システムをロックするかどうかを表示するには、このボックスを選択します。画面を最新表示すると、ロックが解除されます。

メッセージ表示待機標識

メッセージ表示待機標識を表示する場合は、このボックスを選択します。これは 5250 のみのフィーチャーで、これによって、新規メッセージがユーザーを待機中の場合は OIA で標識が表示されま

す。ユーザーが 5250 ホストの「メッセージの表示」パネルにアクセスすると、標識はクリアされます (別の新規メッセージを受信するまで)。

注:

1. ユーザーは、自分のメッセージ・キューの送達タイプを *NOTIFY に設定して、メッセージ表示待機標識を受信する必要があります。
2. このボックスを選択しても、3270 または 3270E の接続には無効です。

非同期更新

非同期更新が使用可能かどうかを表示する場合は、このボックスを選択します。

入力モード

上書きモードが有効かどうか (ブラウザーでサポートされる場合) を表示するには、このボックスを選択します。

絶対的カーソル位置

ホスト・カーソル位置を表示させるには、このボックスを選択します。

カーソル行および列

ホスト・カーソルの行および列位置を表示する場合は、このボックスを選択します。

自動フィールド進行

自動フィールド進行が有効かどうか (ブラウザーでサポートされる場合) を表示する場合は、このボックスを選択します。

フィールド・データ

外部フィールド・データ (「数値のみ」や「フィールドの終了が必要」など) を表示する場合は、このボックスを選択します。

先行入力フィールド **RCP のみ**

先行入力フィールドを表示する場合は、このボックスを選択します。このフィールドは、ユーザーの入力時に先行入力データを表示しますが、フィールドを直接編集することはできません。この設定は、「**先行入力サポートを使用可能にする (Enable type-ahead support)**」設定が選択されている場合のみ構成可能です。 147 ページの『先行入力サポートを使用可能にする』を参照してください。

表示域のレイアウト **Web のみ**

水平 (ページ下部に) OIA を水平に表示する場合は、このボタンを選択します。

垂直 (ページのサイド・フレームに) OIA を垂直に表示する場合は、このボタンを選択します。

スタイル・クラス **Web のみ**

OIA の外観を制御する Cascading Stylesheet (CSS) クラス名を入力します。

DBCS

デフォルト接続に DBCS コード・ページが指定されている場合、プロジェクトに DBCS オプションを構成できます。「レンダリング」タブの DBCS 設定について詳しくは、544 ページの『「レンダリング」タブ』を参照してください。

イベント

「**イベント**」タブには、アプリケーション・イベントのほか、プロジェクトに含まれる優先順位付けされた画面イベント (画面カスタマイズおよび画面組み合わせ) のリストが表示されます。

画面イベントの名前の横にあるチェック・ボックスが選択されている場合、そのイベントはプロジェクトに対して使用可能になっています。画面イベントが使用可能になっていて、かつ、画面認識基準がホスト画面に一致した場合、HATS はそのイベントに指定されているアクションを実行します。画面イベントについての一致検査はそれ以上行われません。画面イベントが使用不可になっている場合、HATS はイベントを無視します。特定の画面イベントをテストする場合、その他の画面イベントは使用不可にすることができます。画面イベントを使用不可にするには、テスト中にチェック・ボックスをクリアします。

HATS アプリケーションでは、使用可能になっている画面イベントのリストに対して、各着信ホスト画面を照らし合わせて検査します。指定の画面に一致する画面イベントが複数ある場合は、画面に一致する最初の画面イベントが適用されます。優先順位が高い画面イベントほど、リストの上部にあります。例えば、いくつかの特定の画面を認識する画面イベントと、より一般的な画面セットを認識する画面イベントがあるとします。リスト内で、後者の画面イベントが前者の画面イベントより上にある場合、画面は、より具体的な基準で認識されて前者の画面イベントに関連するアクションが実行されるのではなく、より一般的な画面認識基準で認識されて関連するアクションが実行されます。

画面イベントの優先順位を変更する場合は、対象となるイベントをクリックして強調表示します。「**上へ**」または「**下へ**」をクリックして、そのイベントをリスト内で上下に移動させます。詳しくは、『画面イベントの優先順位』を参照してください。

注: 特定の次画面を指定する画面イベントがある場合は、その特定の画面が優先されます。詳しくは、197 ページの『次の画面』を参照してください。

アプリケーション・イベントを編集する場合は、「**アプリケーション・イベント**」の下で、編集するイベントのリンクをクリックします。詳しくは、125 ページの『アプリケーション・イベント』を参照してください。

画面イベントの優先順位

プロジェクト・エディターの「**イベント**」タブを使用して、画面イベントの順序を変更できます。詳細については、『イベント』を参照してください。

HATS アプリケーションの実行中に新規ホスト画面が検出されると、HATS はイベント優先順位リストにある最初の使用可能な画面イベントをチェックして、画面認識基準がホスト画面に一致するかどうかを判断します。一致した場合、画面イベン

トについてのそれ以上の一致検査は行われず、最初の画面イベントのアクションが実行されます。一致しなかった場合は、リスト内の次の画面イベントをチェックして、画面認識基準がホスト画面に一致するかどうかが判断されます。このようにして、リスト内の最後の画面イベントまでチェックを続行します。

現在のホスト画面に一致する画面認識基準が画面イベントに存在しない場合、HATS にアクションが設定されていて画面がブランクであると、HATS はブランク画面イベントを処理します。それ以外の場合、HATS は一致しない画面イベントを処理します。 HATS の一致しない画面イベントは、ホスト画面に一致する定義済みの画面イベントがない場合にのみ実行される特殊な画面イベントです。このイベントのデフォルト・アクションとして、デフォルト・テンプレートを適用するホスト画面 (デフォルト変換) が表示されます。

ホスト画面がどの画面イベントにも一致しない場合に実行するアクションは変更することができます。例えば、ページが見つからなかったことをユーザーに通知し、ユーザーがその画面に到達した方法についての情報を収集する Web ページを作成することができます。「**URL を表示**」アクションを使用して、Web ページを表示できます。

一致しない画面イベントのアクションを変更する場合は、「**イベント**」タブに移動し、「**アプリケーション・イベント**」セクションの「**一致しない画面**」をクリックして、エディターを開きます。エディターの下部にある「**アクション**」タブをクリックしてアクションを表示します。「**追加**」、「**編集**」、または「**除去**」をクリックして、unmatchedScreen.evnt ファイルをカスタマイズします。

アプリケーション・イベント

HATS が使用する規則ベースのアプローチにより、画面認識イベントに特定のアクションを適用して画面をカスタマイズできます。また、アクションを HATS イベントに関連付けることもできます (アプリケーション・イベントと呼びます)。アプリケーション・イベントには、ホスト・サーバーとの接続および切断、HATS アプリケーションの始動または停止、アプリケーションがエラーまたは認識されていないホスト画面を検出した場合などのアプリケーションのイベントが含まれます。これらのアプリケーション・イベントを構成すると、これらの事態が発生した場合に特定のアクションを実行できます。

アプリケーション・イベントにアクセスするには、「**HATS Projects**」ビューにアクセスして HATS プロジェクトの「**プロジェクト設定**」をダブルクリックし、「**イベント**」タブを選択します。「アプリケーション・イベント」セクションの「**イベント**」タブで、イベントへのリンクをクリックして該当するエディターを開きます。

各アプリケーション・イベントには、さまざまなアクションが関連付けられています。

以下のリストに、HATS アプリケーション・イベントの説明、およびそのイベントと関連するアクションが発生する場合を示します。

開始　開始イベントは、HATS アプリケーションを始動するときに発生します。開始イベントにデフォルトのアクションはありません。

開始イベントには、次のアクションを指定できます。

- ビジネス・ロジックを実行
- グローバル変数を設定
- グローバル変数を除去
- URL を表示 Web のみ
- URL または SWT コンポジットを表示 RCP のみ
- URL に転送 Web のみ
- マクロ・トランザクションを実行
- 一時停止

接続 接続イベントは、HATS アプリケーションがホスト・サーバーに接続するときに発生します。例えば、接続時にマクロを実行して特定の画面をバイパスし、ログオン画面を表示するようにプロジェクトを構成できます。接続イベントのデフォルトのアクションは、接続の取得です。接続イベントのアクションは、常に開始イベントの後に実行されます。

接続イベントには、以下のアクションを指定できます。

- デフォルト接続を取得
- ビジネス・ロジックを実行
- グローバル変数を設定
- グローバル変数を除去
- URL を表示 Web のみ
- URL または SWT コンポジットを表示 RCP のみ
- URL に転送 Web のみ
- マクロを実行
- マクロ・トランザクションを実行
- 切断
- 一時停止

ブランク画面

ブランク画面イベントでは、ホスト画面がブランクの場合にデフォルト接続で実行するアクションを指定できます。デフォルトでは、このオプションのイベントにデフォルト・アクションが設定されておらず、このイベントは無視されます。アプリケーションで頻繁にブランク画面が表示され、ユーザーが「最新表示」ボタンを押す必要がある場合は、このイベントに「一時停止」アクションを追加して、ホストがホスト画面の描画を完了するまでの待機時間を延ばすこともできます。このイベントにアクションを指定した場合、画面イベント優先順位リストで指定した画面イベント・リストの終わりに達した後、一致しない画面イベントを実行する直前にこのイベントが検査されます。

ブランク画面イベントには、次のアクションを指定できます。

- ビジネス・ロジックを実行
- グローバル変数を設定
- グローバル変数を除去
- URL を表示 Web のみ
- URL または SWT コンポジットを表示 RCP のみ

- URL に転送 Web のみ
- マクロを実行
- マクロ・トランザクションを実行
- 切断
- 一時停止

ホストに最初に接続したときに受信したブランク画面の処理方法を指定できる特殊な設定もあります。詳しくは、 165 ページの『画面処理』を参照してください。

一致しない画面

一致しない画面イベントは、HATS が、HATS アプリケーションのどの画面イベントにも一致しない画面をホスト・アプリケーションから受信したときに発生します。一部の画面にデフォルト・レンダリングを使用するように HATS アプリケーションを構成した場合は、これは通常の事象です。画面イベントで考えられるすべてのホスト画面の認識作業を終了した後には、一致しない画面イベントを構成し、ユーザーに予期しない画面に到達したことを通知して、その画面への到達方法を尋ね、適切な画面に戻るパスを示す URL を表示するようにすることができます。一致しない画面イベントのデフォルトのアクションは、デフォルト変換の表示です。

一致しない画面イベントには、次のアクションを指定できます。

- 変換を適用
- ビジネス・ロジックを実行
- グローバル変数を抽出
- データを挿入
- グローバル変数を設定
- グローバル変数を除去
- URL を表示 Web のみ
- URL または SWT コンポジットを表示 RCP のみ
- URL に転送 Web のみ
- マクロを実行
- マクロ・トランザクションを実行
- キーを送信
- 切断
- 一時停止

エラー エラー・イベントは、HATS がアプリケーション・エラーまたはホスト・エラーを検出したときに発生します。エラー・イベントが発生した場合に GUI でエラーを表示するアクションを追加することもできます。エラー・イベントのデフォルト・アクションでは、 URL (Web アプリケーションの場合は error.jsp) またはコンポジット (リッチ・クライアント・アプリケーションの場合は CustomErrorComposite) が表示されます。

エラー・イベントには、次のアクションを指定できます。

- ビジネス・ロジックを実行
- グローバル変数を設定

- グローバル変数を除去
- URL を表示 Web のみ
- URL または SWT コンポジットを表示 RCP のみ
- URL に転送 Web のみ
- マクロを実行
- マクロ・トランザクションを実行
- 一時停止

切断 切断イベントは、HATS アプリケーションがホスト・サーバーを切断したときに発生します。切断イベントのデフォルトのアクションは、「デフォルト接続を解放」です。

切断イベントには、以下のアクションを指定できます。

- デフォルト接続を解放
- ビジネス・ロジックを実行
- グローバル変数を設定
- グローバル変数を除去
- URL を表示 Web のみ
- URL または SWT コンポジットを表示 RCP のみ
- URL に転送 Web のみ
- マクロを実行

  注: マクロを切断イベントの一部として実行中にタイムアウトになる場合の対処法については、後述する「停止」イベントの説明を参照してください。
- マクロ・トランザクションを実行
- 一時停止

停止 停止イベントは、HATS アプリケーションを停止した後に発生します。停止イベントのデフォルト・アクションは「URL を表示」(Web アプリケーションの場合は stop.jsp)、または「コンポジットを表示」(リッチ・クライアント・アプリケーションの場合は CustomDisconnectComposite) です。

停止イベントには、次のアクションを指定できます。

- ビジネス・ロジックを実行
- グローバル変数を設定
- グローバル変数を除去
- URL を表示 Web のみ
- URL または SWT コンポジットを表示 RCP のみ
- URL に転送 Web のみ
- マクロを実行
- マクロ・トランザクションを実行
- 一時停止

注: マクロを切断イベントの一部として実行中にタイムアウトになって完了しない場合は、切断イベントは完了せず、HATS 接続は切断されませ

ん。マクロのタイムアウトは、通常は、マクロが未知のホスト画面に遭遇する場合に起こります。この状態になったときに「停止」イベントがホスト接続を正常に終了するように構成するには、以下のステップを実行します。

1. 「**停止**」イベントのリンクをクリックして、該当するエディターを開きます。
2. エディターで、「**ソース**」タブをクリックします。
3. `<release enabled="true"/>` アクションを、`<show/>` アクションの前に追加します。
4. 「停止」イベント・ソースは次の例のようになります。

```
<?xml version="1.0" encoding="UTF-8"?>
<event description="" type="stop">
    <actions>
        <release enabled="true"/>
        <show enabled="true" page="stop.jsp" url="/stop.jsp"/>
    </actions>
</event>
```

5. プロジェクト設定ファイルを保存してアプリケーションを再始動します。

その他

「**その他**」タブには、キーボード・サポート、クライアント・ロケール、接続パラメーター・オーバーライド、非同期更新、グローバル変数オーバーライド、およびクライアント設定の設定が表示されます。

カスタマイズ設定ツリー内で個々のノードをクリックすることにより、該当のデフォルト・プロジェクト設定をカスタマイズすることができます。

キーボード・サポート

キーボード・サポートについては、以下の設定をカスタマイズできます。

キーボード・サポートを使用可能にする

ユーザーが物理的なキーボード・キーを使用してホストと対話できるようにするには、このチェック・ボックスを選択します。これによりユーザーは、ホスト AID キー (F1 キー、SYSREQ キー、RESET キー、ATTN キーなど) にマップされた特定の物理キーを押すことができます。マップされた物理キーボード・キーを使用してブラウザーと対話する場合は、この機能をオフに切り替えることができます。

HATS Web プロジェクトでキーボード・サポートを使用するには、サポートされている Web ブラウザーが必要です。サポートされる Web ブラウザーのリストおよび制限については、「Rational Host Access Transformation Services: System Requirements」(http://www.ibm.com/support/docview.wss?uid=swg27011794) および「Host Access Transformation Services 9.5 - Known issues and workarounds」(http://www.ibm.com/support/docview.wss?uid=swg27046998) を参照してください。

初期キーボード・サポート状態をオン

最初に GUI が表示されるときにキーボードを使用可能にするには、このチェック・ボックスを選択します。

以下のオプションから、キーボード・サポートをブラウザー・ページに表示されるファンクション・キーに限定するかどうかを指定します。

ホスト・キーパッドに表示されるホスト機能だけをサポート (Support only host functions displayed on host keypad) Web のみ

このオプションを選択すると、認識される機能ボタンまたはリンクについてブラウザー・ページがスキャンされます。機能が検出された場合は、それらの機能だけがサポートされます。機能が検出されない場合は、すべての機能がサポートされます。一部の旧バージョンの HATS では、これがデフォルトの振る舞いでした。これらのバージョンからマイグレーションしたアプリケーションは、以前と同様に機能します。

すべてのマップ済みホスト機能のサポート (Support all mapped host functions) Web のみ

ブラウザー・ページに存在するホスト・キー・ボタンやリンクに関係なく、キーボード・サポートをすべてのマップ済み機能と連動させるには、このオプションを選択します。これは、新規の HATS アプリケーションのデフォルトの動作です。

クライアント・ロケール

ボタンの表題とメッセージは、さまざまな言語で表示できます。表示する言語を獲得するには、以下のクライアント・ロケール設定をカスタマイズします。

クライアントのデフォルト・ロケールから (From the client's default locale)

Web アプリケーションの場合、ボタンの表題およびメッセージを表示するのに使用される言語は、ユーザーのブラウザーで指定されている言語によって決定します。リッチ・クライアント・アプリケーションでは、デフォルト・オペレーティング・システムの値、あるいはリッチ・クライアント環境の始動時に指定した値のいずれかによって、言語が決まります。

サーバーの基本ロケールから Web のみ

ボタンの表題およびメッセージを表示するのに使用される言語は、アプリケーションが配置されているサーバーのロケールによって決定します。

常に以下の言語を使用

ボタンの表題およびメッセージに使用する言語を、ドロップダウン・リストから選択できます。

接続パラメーターのオーバーライド

HATS では、HATS ホスト接続の接続パラメーターとして Host On-Demand セッション・パラメーターを使用することができます。このパラメーターの詳細については、Host On-Demand Knowledge Center (http://publib.boulder.ibm.com/infocenter/hodhelp/v11r0/index.jsp) にアクセスして「session parameters」で検索してください。これらのパラメーターに関する情報は、http://www-01.ibm.com/support/docview.wss?uid=swg21251884 にある資料でも入手できます。

HATS Web アプリケーションの場合、これらの接続パラメーターは、HATS プログラムへのアクセスに使用する URL 上でオーバーライドできます。例えば、以下のような URL を入力できます。

```
http://servername:port/appname/index.jsp?host=9.42.141.26&sessionType=1&port=523&
LUName=LU00001&TNEnhanced=true
```

サーバー名、ポート、および HATS アプリケーション名に続けて開始ページの index.jsp を指定し、その後に疑問符 (?) を付けて、各パラメーターの名前/値の組 (name=value) をアンパーサンド (&) で区切って指定します。これらの値は、application.hap ファイルの他の場所で使用されている値ではなく、Host On-Demand によって定義されたテキスト値です。例えば、3270 の場合は sessionType=1、5250 の場合は sessionType=2 になります。

Host On-Demand セッション・パラメーターに加え、 SSLP12FilePath パラメーターおよび SSLP12Password パラメーターもオーバーライド可能です。 SSLP12FilePath パラメーターおよび SSLP12Password パラメーターでは、SSL 接続に使用される PKCS12 鍵ストア・ファイルの場所とパスワードを指定します。詳しくは、 166 ページの『セキュリティー』を参照してください。

リッチ・クライアント・アプリケーションのユーザーが、「アプリケーション」ビューのポップアップ・メニューにある「**プロパティー**」項目を使用して接続パラメーターをオーバーライドする方法については、 95 ページの『接続パラメーター』を参照してください。接続パラメーターのオーバーライドを使用してリッチ・クライアント・アプリケーション・インスタンスをプログラマチックに開始する方法については、「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

接続パラメーターがある場合に、そのうちのどれをオーバーライド可能にするかを制御するには、以下のいずれかのラジオ・ボタンを選択します。

すべてのデフォルト接続パラメーターをクライアント要求でオーバーライドできるようにする (Allow all default connection parameters to be overridden by client requests)

このボタンを選択すると、テーブル内でリストおよび選択されている接続パラメーターを除くすべての接続パラメーターをオーバーライドできるようになります。

すべてのデフォルト接続パラメーターをクライアント要求でオーバーライドできないようにする (Do not allow any default connection parameters to be overridden by client requests)

このボタンを選択すると、テーブル内でリストおよび選択されている接続パラメーターを除くすべての接続パラメーターをオーバーライドできなくなります。

接続パラメーターを追加または除去することによって、選択内容に例外を設けることができます。使用可能な接続パラメーターのリストを取得するには、「**追加**」ボタンをクリックします。

パラメーターのオーバーライド時に使用する接続のタイプの選択

実行時に接続パラメーターがオーバーライドされると、HATS はそのオーバーライド値を使用して、アプリケーションの作成時に定義されたものとは異なる変換接続

を作成します。 HATS がこの新規の変換接続を作成するためのテンプレートとしてどの接続を選択するかは、**connectionName** 接続パラメーターを指定したかどうかによって決まります。

- **connectionName** パラメーターを指定しなかった場合、HATS は、HATS アプリケーションのデフォルト接続を基に基本変換接続を作成します。
- **connectionName** パラメーターは指定したが、その値が有効な接続名でない場合、HATS は、HATS アプリケーションのデフォルト接続を基に基本変換接続を作成します。
- **connectionName** パラメーターの値としてバックグラウンド接続の名前を指定した場合、HATS は、指定されたバックグラウンド接続を基に完全変換接続を作成します。
- **connectionName** パラメーターの値として HATS アプリケーションのデフォルト接続の名前を指定した場合 (通常は `main`)、HATS は HATS アプリケーションのデフォルト接続を基に完全変換接続を作成します。

HATS が基本変換接続を作成する場合は、一部の機能が無効になります (テンプレート接続として選択された接続で有効になっている場合も含む)。基本接続ではプールが常に無効になります。また、接続マクロおよび切断マクロと同様、接続タイムアウトおよび切断タイムアウトが無視されます。テンプレート接続の HOD 接続パラメーター (ホスト名など) およびポートがテンプレート接続からコピーされ、接続パラメーターに対して指定されているすべてのオーバーライドが適用されて、新規の基本変換接続が作成されます。基本接続は常にアプリケーションのデフォルト変換接続を基にしていることに注意してください。

HATS が完全変換接続を作成する場合は、**connectionName** パラメーターで指定された接続に関連付けられているすべての接続設定 (プール設定、接続マクロおよび切断マクロ、接続タイムアウトおよび切断タイムアウトなど) が使用されます。その後、接続パラメーターのオーバーライドがある場合にはそれらが適用されて、新規の変換接続の作成に使用されるプール仕様が作成されます。

connectionName パラメーターの値としてバックグラウンド接続の名前を指定し、その他のオーバーライドは指定しなかった場合は、事実上、そのアプリケーション・インスタンスの変換接続として使用する接続を切り替えていることになります。

すべての HATS 接続は、プール仕様に基づいています。プール仕様は、その仕様を基に作成されるすべての接続のプロパティーを記述した内部オブジェクトです。実際には HATS Toolkit でプール仕様を作成し、ランタイムではこれらの接続ファイル (main.hco など) を使用して、実際の Telnet 接続を実行時に作成します。接続パラメーターのオーバーライドを指定した場合、HATS ランタイムは新規接続のプロパティーを記述した新規のプール仕様オブジェクトを、必要に応じて動的に作成します。プール仕様は、Web アプリケーションの HATS 管理パネルで参照することができます。プールが無効になっている場合、パラメーターのオーバーライド指定があるために動的に作成されたプール仕様は、**connectionName** パラメーターが指定されているかどうかとは無関係に、最後のアクティブ接続が終了するとともに自動的に破棄されます。

接続パラメーターのオーバーライドは、HATS アプリケーションにアクセスするユーザーごとに別々に指定することができます。例えば、各ユーザーがそれぞれ別の LUName オーバーライドを指定している場合、HATS ランタイムはユーザーごとに

異なるプール仕様を作成します。プール仕様が自動的に破棄されるようにするには、元の接続に対してプールを無効にしてください。複数のユーザーが全く同じオーバーライドを指定した場合はプール仕様が 1 つだけ作成され、それを再利用することで、同一構成を持つ多数の Telnet 接続が作成されます。例えば、複数のユーザーが所属する部門全体で、ある特定のホスト名のオーバーライドが指定されている場合は、新規のプール仕様が 1 つだけ作成され、その仕様を使用する接続がなくなった時点で破棄されます。

connectionName パラメーターを接続パラメーターのオーバーライドとして指定するには、プロジェクト設定エディターの「接続パラメーターのオーバーライド」ページにある「**その他**」タブで、このパラメーターをオーバーライドに対して有効にしておく必要があります。

disconnectOnClose パラメーター

HATS では、セッションが切断される前にブラウザーが閉じた場合にホストを自動的に切断する自動切断機能が提供されます。詳しくは、134 ページの『自動切断および最新表示 Web のみ』を参照してください。自動切断機能を使用しないアプリケーションでは、disconnectOnClose パラメーターによって同様の機能が実行されます。このパラメーターによって、ユーザーがセッションをホストから手動で切断する前にブラウザーが閉じた場合に、HATS サーバーがホストの Telnet セッションをすぐにリセットできるようになります。ユーザーが「切断」をクリックすると、通常は、切断されます。Telnet セッションのリセットは、元のブラウザーが閉じた場合に新しいブラウザーを起動する隠しフレームをブラウザー内に設定することによって実現されます。新規のブラウザー・ウィンドウは、切断信号を送信してから、閉じます。

この機能は、`disconnectOnClose=true` が指定されていて、かつ、サポートされているバージョンの Internet Explorer を使用している場合にのみ有効になります。disconnectOnClose パラメーターの指定には、大/小文字の区別があります。サンプル URL 要求は、以下のとおりです。

```
http://hostname/appname/index.jsp?disconnectOnClose=true
```

appname は、HATS アプリケーションの名前です。

disconnectOnClose parameter パラメーターを使用する場合、以下の条件は真です。

- HATS セッション・フレーム経由ですべてのナビゲーションを実行する場合 (ページ内のリンクのクリックを含む)、disconnectOnClose パラメーターが機能します。
- ユーザーがアドレス・バーまたはお気に入りの項目を使用してブラウザーに情報を表示する場合、disconnectOnClose パラメーターは機能しません。ユーザーがこれらの手段でブラウザーに情報を表示し、ブラウザーを閉じる前に HATS アプリケーションに戻らない場合は、自動切断が発生しません。
- フレームのサポートが使用不可になっている場合、またはポップアップ防止ソフトウェアがインストールされている場合は、disconnectOnClose パラメーターが機能しません。
- ポートレット環境では HTML フレームを使用するため、disconnectOnClose パラメーターは機能しません。

非同期更新 RCP のみ

HATS リッチ・クライアント・アプリケーションの場合、非同期更新機能はデフォルトで有効になっています。この機能を無効にするには、「**非同期更新を使用可能にする**」ボックスをクリアします。非同期更新メカニズムの主な目的は、ホストが新しい画面を送信したために基本のホスト画面が変更された場合に、GUI 表示を自動的に更新することです。これには以下の利点があります。

- リッチ・クライアント・アプリケーションのユーザーに対する応答時間の短縮。
- リッチ・クライアント・インターフェースのエミュレーターとの類似性。リッチ・クライアント環境の画面最新表示は、エミュレーターの画面最新表示と同じように行われます。

自動切断および最新表示 Web のみ

HATS Web アプリケーションの場合、自動切断および最新表示機能は、AJAX 実装を使用するクライアント・プル・メソッド、またはアプレット実装を使用するサーバー・プッシュ・メソッドのいずれかで実行できます。 AJAX 実装を使用可能にするには、「**クライアント・プル (AJAX) を使用可能にする**」オプションを選択します。アプレット実装を使用可能にするには、「**サーバー・プッシュ (アプレット) を使用可能にする**」オプションを選択します。これらの機能を両方とも使用不可にするには、「**使用不可**」を選択します。デフォルトは「**使用不可**」です。

クライアント・プル (AJAX) メソッドの使用

HATS Web アプリケーションの場合は、クライアント・プル (AJAX) 実装メソッドを構成できます。これには、自動切断機能と自動最新表示機能が含まれます。

自動切断機能は、事前に定義された期間ブラウザーが HATS アプリケーションと接触していない場合を検出する機能を提供します。検出されると、ブラウザー・セッションに関連付けられたリソースをクリーンアップしてサーバーの負荷を減らすために、切断処理が実行されます。これを行うために、切断イベントが実行され、切断マクロ (構成されている場合) が実行されます。

自動最新表示機能の目的は、ホストが新しい画面を送信したために基本のホスト画面が変更された場合に、HATS アプリケーションの Web ページ表示を自動的に最新表示することです。これにより、ホストから非同期で変更された画面を表示するためにユーザーが「最新表示」ボタンをクリックする必要はなくなります。

自動最新表示を使用可能にすると、以下の場合には Web ページ表示は自動的には更新されません。

- ユーザーが Web ページ表示を更新したが、変更を送信していない。
- 特定の画面について自動最新表示が使用不可に設定されている。以下の行を画面変換ファイルの </HATS:Form> タグの直後に追加することにより、特定の画面について自動最新表示を使用不可に設定できます。

```
<script>
disableBrowserRefresh(formID);
</script>
```

サーバー・プッシュ (アプレット) 実装メソッドとは異なり、クライアント・プル (AJAX) メソッドでは、特定のブラウザーを使用する必要はなく、HATS によってサポートされるすべてのブラウザーを使用できます。また、サーバー・プッシュ

(アプレット) メソッドとは異なり、サーバーへの追加の接続は不要です。そのため、HATS 管理者にとって構成が簡略化されます。クライアント・プル (AJAX) メソッドとサーバー・プッシュ (アプレット) メソッドは相互に排他的です。いずれかのクライアント・プル (AJAX) 機能が有効になっている場合、サーバー・プッシュ (アプレット) 構成は無視されます。

注:

1. クライアント・プル (AJAX) メソッド (自動切断機能および自動最新表示機能を含む) は、JSR 286 ポートレットを含む HATS Web 画面変換アプリケーションに対してのみサポートされます。
2. このメソッドと以下の HATS 機能の併用はサポートされません。
   - JSR 168 ポートレット・プロジェクト
   - WebFacing プロジェクト
   - リンク HATS/WebFacing プロジェクト
   - 統合オブジェクト
   - EJB プロジェクト
   - Web サービス
   - リッチ・クライアント・プロジェクト
3. モバイル装置用に最適化された HATS Web アプリケーションは、キーボード・ホスト・キー・サポートを提供しません。このため、新しいホスト画面がホスト・アプリケーションから非同期で受信されると、ユーザーが HATS アプリケーションの GUI ビューに入力したデータを自動最新表示サポートが上書きします。
4. iPhone または iPod touch デバイスで実行されているブラウザーから別のアプリケーションに切り替えると、ブラウザーからの AJAX ポーリングが停止します。この結果、HATS にアクセスしているブラウザーから、デバイス上の別のアプリケーションに切り替えたとき、**切断を待機する時間 (秒)** 間隔が経過した後、HATS によりブラウザー・セッションが切断されます。この間隔を長くするのが適切な場合、あるいは iPhone または iPod touch デバイスからアクセスされる HATS アプリケーションの自動切断機能を無効にするのが適切な場合があります。詳しくは、『クライアント・プル (AJAX) の設定』を参照してください。

クライアント・プル (AJAX) の設定:

自動切断機能および自動最新表示機能を構成するには、以下の設定を使用します。

ポーリング間隔 (秒)

ブラウザーが HATS アプリケーションをポーリングして、切断タイマー (使用可能な場合) を再始動し、ホスト画面の更新を確認する、秒単位の間隔。値は、秒単位で、小数点以下第 1 位まで指定します。最小値は 1.0 秒です。デフォルトは **5.0** 秒です。 「**切断**」が選択されている場合、ポーリング間隔値は「**切断を待機する時間 (秒)**」設定の値より少なくとも 1 秒小さい必要があります。

注: アプリケーション・サーバーの WebSphere Application Server「**セッション管理**」設定で「**タイムアウトの設定**」オプションを選択することにより HTTP セッション・アイドル・タイムアウトが構成されている場

合は、HATS アプリケーションのブラウザー・ポーリングは効果的に HTTP セッション・アイドル・タイムアウト機能を使用不可にします。このため、HATS ランタイムが責任を持って HTTP セッション・アイドル・タイムアウト期間をモニターし、アイドル時間が経過する前にユーザー・アクティビティーがなければ、切断イベントの実行を含め、HATS セッションの切断を開始します。

自動切断および自動最新表示の両方の機能が現行ページで使用不可に設定されている場合、HTTP セッション・アイドル・タイムアウトが起動されると、HATS はセッションを切断し、切断イベントを実行しません。

切断 自動切断機能を使用可能にするには、このオプションを選択します。選択されている場合、「**切断を待機する時間 (秒)**」フィールドに指定された時間内にクライアントが HATS アプリケーションをポーリングしないと、HATS アプリケーションは切断処理を開始します。

注: ブラウザー・セッションに関連付けられたリソースをクリーンアップしてサーバーの負荷を減らすために、切断処理が実行されます。これを行うために、切断イベントが実行され、切断マクロ (構成されている場合) が実行されます。

切断を待機する時間 (秒)
自動切断機能を実行するまでに待機する秒単位での時間。値は、秒単位で、小数点以下第 1 位まで指定します。最小値は 2.0 秒です。デフォルトは **15.0** 秒です。

最新表示
自動最新表示機能を使用可能にするには、このオプションを選択します。選択されている場合、ユーザー入力がなく、ホスト画面が変更されたことをポーリング応答が示していると、ブラウザーは最新表示を開始します。

サーバー・プッシュ (アプレット) メソッドの使用

注: サーバー・プッシュ (アプレット) メソッドは推奨されません。このメソッドは現在はサポートされていますが、IBM は将来のリリースでこの機能を除去する権利を持っています。代わりに、クライアント・プル (AJAX) メソッドの使用をお勧めします。

サーバー・プッシュ (アプレット) 実装メソッド (非同期更新アプレットとも呼ばれる) によって実行される機能は 2 つあります。

- 非同期ホスト更新をブラウザーにプッシュする (Pushing asynchronous host updates to the browser)
  - ユーザーが「**最新表示**」ボタンをクリックしなくても、ホスト更新が自動的に表示されます。
- ブラウザーのクローズを HATS サーバーに通知する (Notification of browser close to HATS server)
  - この機能によって、HATS サーバーでブラウザー・セッションに関連付けられたリソースをクリーンアップしてサーバーの負荷を減らすことができます。

非同期更新アプレットを使用すると、 HATS サーバーからユーザーのブラウザーに変換済みの端末画面を送信するのに使用される接続に加えて、クライアントで稼働するアプレットと HATS サーバーの間に別にソケット接続が確立されます。このソケット接続は、クライアントとサーバー間で容量の小さいメッセージを受け渡しする場合に使用されます。容量の小さいメッセージとは、ホストからの非同期の画面最新表示や、HATS サーバーに対するブラウザーの終了通知に関するものです。このソケット接続のトラフィック量は、通常は非常に少ないものです。このソケット接続をセットアップすると、次のような一連のイベントが発生します。

1. ユーザーがブラウザーで HATS アプリケーションの URL にアクセスします (リンクを使用するか、アドレス・バーに URL を入力します)。
2. HATS サーバーが要求を受信し、ブラウザーに応答を送信します。この応答には非同期更新アプレットの Java コードが含まれています。
3. クライアントがアプレットをダウンロードすると、アプレットは個別のソケット接続を開設して、 HATS サーバーと直接通信しようとします。このソケット接続に使用するクライアント側のローカル・ポートは、使用可能で未使用のポートからランダムに選択され、構成することはできません。アプレットの接続先となるリモート HATS サーバーの IP 名およびポート情報は、アプレットがクライアントにダウンロードされるときにサーバーから提供されます。サーバー側の HATS アプリケーション構成でそのサーバーのポート番号に値 0 が指定されていると、ポート番号が 1024 より大きい未使用のポートが HATS サーバー上でランダムに選択され、パラメーターとしてアプレットに提供されます。サーバーで選択されたポート番号は、ブラウザーで HATS 変換のページ・ソースを表示すると確認できます。ページ・ソースを検索して、`PARAM NAME="port"` に指定されている値を調べてください。アプレット・パラメーター設定の詳細については、『サーバー・プッシュ (アプレット) の設定』を参照してください。

アプレットを実行するには、Java 対応ブラウザーが必要です。サポートされるブラウザーの最新リストについては、「Rational Host Access Transformation Services: System Requirements」(http://www.ibm.com/support/docview.wss?uid=swg27011794) を参照してください。

注: このアプレットは、HATS ポートレットでは機能しません。詳しくは、 457 ページの『第 19 章 WebSphere Portal と HATS』を参照してください。

サーバー・プッシュ (アプレット) の設定:

サーバー・プッシュ (アプレット) 実装メソッドを構成するには、以下の設定を使用します。

ホスト名

これは、HATS アプリケーションを実行するサーバーのホスト名または IP アドレスです。サーバーが水平クラスター化された環境にある場合、application.hap ファイルをサーバーで展開した後にこのアプリケーション・ファイルを編集し、ホスト名を変更する必要があります。

ブラウザー側のアプレットでは、TCP/IP ソケットを使用して HATS サーバーと通信します。ソケットを作成するには、HATS アプリケーションをホスティングしているサーバーのホスト名が必要です。HATS がブラウザーの URL によって指定された同じ Web サーバーに配置されている場合、ホスト名パラメーターを指定する必要はありません。アプレットはその Web サ

ーバーの IP アドレスを使用してソケット接続を確立します。 HATS アプリケーションが別のサーバーにある場合、ホスト名パラメーターは、ホスト名または IP アドレスを使用して指定する必要があります。

サーバー・ソケットのポート

サーバー上のポート。クライアントがこのサーバーと通信するには、このポートを使用する必要があります。ゼロ (0) は、1024 より大きい値のランダムな未使用ポートが使用されることを表します。2043–4544 のようにポートの範囲を指定することもできます。また、3045, 1345, 9596 のようにコンマ (,) で区切ったポート・リストを指定することもできます。

注: HATS が垂直クラスターになっている (同一の物理サーバー上に複数の HATS インスタンスがある) 場合は、HATS インスタンスごとに別々のポートを使用しなければなりません。ゼロ (0) 以外の値を使用する場合は、少なくとも HATS インスタンスと同じ数のポートを指定する必要があります。各 HATS インスタンスはその始動時に、空いているポートが見つかるまで指定のポート群を試行していきます。

切断を待機する時間 (秒)

ブラウザーが閉じてからユーザーが切断されるまでにアプリケーションが待機する時間 (秒)。ブラウザーからしばらく離れたときに確実にアプリケーションを切断する場合にも、使用できます。値は、秒単位で、小数点以下第 1 位まで指定します。デフォルトは、**300.0** 秒 (5 分) です。有効な値は 300.0 秒以上の数値です。

アプレットには、ブラウザー固有の設定と Windows 以外のマスクの設定を追加で指定できます。これらの設定を変更するには、application.hap ファイルを編集します。

新規プロジェクトを作成する場合は、<classSettings> タグの Web Content/WEB-INF/profiles/application.hap ファイルにタグが生成されます。アプレットのデフォルト設定は次のとおりです。

```
<class name="com.ibm.hats.common.AppletSettings">
   <setting name="disconnectDelay" value="300000"/>
   <setting name="enable" value="false"/>
   <setting name="hostname" value=""/>
   <setting name="ie" value="disconnect|refresh"/>
   <setting name="nonWindows" value="disconnect|refresh"/>
   <setting name="other" value="disconnect|refresh"/>
   <setting name="port" value="0"/>
</class>
```

各パラメーターの概要を以下に示します。

disconnectDelay

このパラメーターは、**「サーバー・プッシュ (アプレット) を使用可能にする」**設定の GUI で**「切断を待機する時間 (秒)」**設定から設定されます。ブラウザーを閉じたり、ブラウザーで新しいページをロードしたりするたびに (つまり、ユーザーが HATS アプリケーションを表示しなくなるたびに)、アプレットは、この時間だけ待機した後にホストからユーザーを切断します。値が小さすぎると、アプリケーションでページ間を移動する場合に、ユーザーが切断されることがあります。この値を高く設定しても実害はありま

せんが、ユーザーがブラウザーをクローズしても、この時間が経過するまではサーバーおよびホストのリソースが保持されます。デフォルト値は、300000 ミリ秒 (5 分) です。

enable この設定を使用して、どの自動切断および最新表示更新方式を使用するかを構成します。クライアント・プル (AJAX) メソッドを構成するには AJAX を指定し、サーバー・プッシュ (アプレット) メソッドを構成するには true を指定し、両方のメソッドを使用不可にするには false を指定します。デフォルトは **false** です。true が指定されると、指定されたポートで listen ソケットが確立されます。このアプリケーションに接続する各ブラウザーでは、そのページにアプレット・タグが挿入されます。

hostname

アプレットが HATS サーバーに再度接続できるようにするには、application.hap ファイルで HATS サーバーのホスト名または IP アドレスを構成する必要があります。

注: 水平クラスタリングでは、application.hap ファイルを編集してホスト名を変更する必要があります。.hap ファイルの手動編集は、WebSphere Application Server 管理を使用して (サーバー上に) .ear が展開された後に行う必要があります。

port クライアントがそのサーバーと通信する際に使用する必要がある、サーバー上のポートを選択します。ゼロ (0) は 1024 より上のランダムなポートを指定します。ポートの範囲 (例えば、2043-4544) を指定することも、コンマ区切りのポート・リスト (例えば、3045,1345,9596) を指定することもできます。デフォルト値はランダムです。

注:

1. HATS がファイアウォールの後ろにある場合は、値ゼロ (0) を指定しないでください。指定すると、任意のポートがファイアウォールを通過するのが困難になる場合があるためです。
2. 垂直クラスタリングでは、HATS がファイアウォールの後ろにある場合にポート範囲を指定できます。HATS が Web サーバー上にある場合は、ポート番号の代わりにゼロ (0) または範囲を指定できます。どちらの場合も、Web サーバーで個別にポート番号を指定することを強くお勧めします。

以下のブラウザー関連の設定は、特定のブラウザーまたは環境で使用される機能を決定します。

nonWindows

Windows 以外のオペレーティング・システムで実行するブラウザーの設定を指定します。指定可能なオプションは disconnect、refresh、local です。このアルゴリズムは l、d、および r を検索し、分離文字は無視します。オプションを適用するには、オプションがオペレーティング・システムの設定とブラウザーの設定の両方にリストされている必要があります (例えば、nonWindows=dr other=dr)。デフォルト値は **disconnect|refresh** です。

ie すべてのバージョンの Internet Explorer のための設定を指定します。デフォルト値は **disconnect|refresh|local** です。

other その他のブラウザー (例: Opera、Mozilla) で使用する設定を指定します。デフォルト値は **disconnect|refresh** です。

上にリストしたブラウザー関連の設定に対して、以下の値を使用できます。

disconnect(d)
ブラウザーを閉じると、disconnectDelay タイマーの設定時間 (application.hap ファイルで指定) が経過した後に、この設定によって HATS サーバーからホストへのホスト・サイド接続がクリーンアップされます。この機能は、どの Java 対応ブラウザーでも使用できます。

注: Internet Explorer 以外のすべてのブラウザーでは、ユーザーが HATS アプリケーションを表示しなくなると、 disconnectDelay タイマーの設定時間経過後に、この設定によってホスト・サイド接続がクリーンアップされます。Internet Explorer の設定について詳しくは、以下の disconnectAlways を参照してください。

disconnectAlways
このフィーチャーにより、Internet Explorer の非同期更新アプレットの切断動作を、他のすべてのブラウザー (Internet Explorer 以外) と同様にすることができます。両者の振る舞いは次のとおりです。

- Internet Explorer のデフォルトの振る舞い (application.hap ファイルで disconnect を指定した場合): ブラウザーを閉じると、application.hap ファイルで指定した disconnectDelay タイマーの時間が経過した時点で、HATS サーバーは接続をクリーンアップします。ユーザーが表示を切り替えた場合は、disconnectDelay タイマーの値に関係なく、セッションを再び表示して再利用することができます。
- 他のすべてのブラウザーのデフォルトの振る舞い: ブラウザーを閉じても表示を切り替えても、 application.hap ファイルで指定した disconnectDelay タイマーの時間が経過した時点で、 HATS サーバーはセッションをクリーンアップします。

このフィーチャーを使用すれば、Internet Explorer でも、ユーザーがブラウザーを閉じた場合または HATS アプリケーションから表示を切り替えた場合に、 disconnectDelay タイマーで指定した時間が経過するとセッションがクリーンアップされるように指定できます。

Internet Explorer の振る舞いを変更するには、`disconnect` を `disconnectAlways` で置き換えます。すると、Internet Explorer のパラメーター・リストは次のように表示されます。

```
<setting name="ie" value="disconnectAlways|refresh|local"/>
```

.

refresh(r)
新規コンテンツが使用可能になったときにサーバーがページを自動的に更新できるようにします。これは、Java 対応のブラウザーで機能します。

local(l) ページの表示スペースのみを最新表示します。これは、Internet Explorer V7、および Internet Explorer V6 の比較的新しいリリースでのみ正常に機能します。他の環境では JavaScript エラーが発生してブラウザーが強制終了する場合があります。

非同期更新アプレットのクライアント・サイド・トレースを使用可能にするには、HATS エントリー・サーブレットの Web アドレスで、HTML パラメーター appletJavaConsoleTrace=true を指定する必要があります。例えば、以下のようになります。

```
http://myhost/myhats/entry?appletJavaConsoleTrace=true
```

注:

1. これは HATS サーバー・サイド・トレースに対するクライアント・サイド・トレースで、サーバー・サイド・トレースを補完してアプレットの問題をデバッグします。
2. このトレースは、IBM サポート担当員によって要求された場合にのみ使用可能にしてください。

グローバル変数のオーバーライド

この機能を使用して、ユーザーからアプリケーションにデータを渡すことができます。例えば、ユーザーがユーザー ID とパスワードを入力し、それを使用してマクロを実行するようにできます。

HATS リッチ・クライアント・アプリケーションでは、ユーザーは、「アプリケーション」ビューのポップアップ・メニューの「**プロパティー**」項目を使用してグローバル変数をオーバーライドします。詳しくは、96 ページの『変数』を参照してください。グローバル変数のオーバーライドを使用してリッチ・クライアント・アプリケーション・インスタンスをプログラマチックに開始する方法については、「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

HATS Web アプリケーションの場合、ローカル・グローバル変数をオーバーライドするには、HTML パラメーター **hatsgv**_*varName*=*newValue* をクライアントの HTTP 要求に追加します。共用グローバル変数をオーバーライドするには、パラメーター名を **hatssharedgv**_*varName* にします。

ユーザーが UserID という名前の共用グローバル変数の値を作成および設定できるようにするには、変換 JSP の <HATS:Form> タグの本文に以下の記述を追加して、入力フィールドを用意します。

```
<INPUT name="hatssharedgv_UserID" type="TEXT" size="10">
```

ユーザーが入力した値が、共用変数 UserID の新規の値になります。 JavaScript によって値を作成し、ユーザーに新規の値を入力させない場合は、隠し入力フィールドを使用して値を自動的に設定します。

```
<INPUT name="hatssharedgv_UserID" size="10" type="HIDDEN" value="alice">
```

グローバル変数を新規作成したり、URL 内のグローバル変数の値を変更したりするには、GET 要求で次の URL のような URL を使用できます。

```
http://hostname/appname/entry?hatsgv_UserID=bob
```

ユーザーのクライアント要求によって作成または変更できる HATS グローバル変数を制御するには (存在する場合)、以下のラジオ・ボタンの 1 つをクリックします。

すべてのグローバル変数をクライアント要求でオーバーライドできるようにする
このボタンにより、テーブル内のリストされたグローバル変数や選択されたグローバル変数以外のすべてのグローバル変数をオーバーライドできます。

すべてのグローバル変数をクライアント要求でオーバーライドできないようにする
このボタンをクリックした場合は、テーブル内のリストされたグローバル変数や選択されたグローバル変数以外のグローバル変数がオーバーライドできません。

グローバル変数の例外を追加するには、「**追加**」ボタンをクリックします。この画面では、テキスト・ボックスにグローバル変数名を入力するか、ドロップダウン・リストから選択できます。共用可能なグローバル変数であることを示すには、「**共用グローバル変数**」チェック・ボックスを選択します。詳しくは、 399 ページの『第 12 章 グローバル変数との対話』を参照してください。

テーブルのエントリーを編集する場合は、「**編集**」をクリックします。「**除去**」ボタンを押すと、強調表示されたグローバル変数が例外のリストから削除されます。

注: HATS V4 からマイグレーションしたアプリケーションに対するデフォルトでは、すべてのグローバル変数をクライアント要求によってオーバーライドできます。この設定は、アプリケーションをインポートした後にこの画面で変更できます。HATS V5 以降の新規アプリケーションの場合、デフォルトでは、クライアント要求によるグローバル変数のオーバーライドは許可されません。

クライアント設定

次のクライアント設定をカスタマイズできます。

HTTP 圧縮を使用可能にする **Web のみ**
このボックスを選択すると、HTTP 圧縮により、HATS ランタイム (WebSphere Application Server で実行されます) とユーザーのブラウザーの間で転送されるバイト数が削減されます。こうすると、 HATS ランタイムとブラウザー間の転送時間が短くなり (応答時間の改善)、ネットワークを流れるバイト数が少なくなります (ネットワーク使用状況の改善)。

注:

1. 圧縮が正しく機能しているかどうかを判断し、圧縮前と圧縮後のページ・サイズのバイト数を確認するには、runtime.properties ファイル (HATS Toolkit のデバッグ・モードで実行している場合は runtime-debug.properties ファイル) の trace.UTIL 行を更新してトレースを使用可能にします。例: `trace.UTIL=7`。詳しくは、 557 ページの『付録 A. ランタイム・プロパティー・ファイル』を参照してください。トレース・ファイルで、runtime.filters.CompressionFilter というトレース・エントリーを検索します。例えば、次のようになります。

```
+-------------------------------------+
Text UTIL runtime.filters.CompressionFilter.doFilter()
 17.50.10.140 11/27/06 Servlet.Engine.Transports : 1
enable compression: true 0000CBAx-81SRWWVmQfQ8-_47oK:-1

+-------------------------------------+
Text UTIL runtime.filters.CompressionFilter.doFilter()
 17.50.10.140 11/27/06 Servlet.Engine.Transports : 1
size before: 25315 0000CBAx-81SRWWVmQfQ8-_47oK:-1
```

```
+-------------------------------------+
Text UTIL runtime.filters.CompressionFilter.doFilter()
 17.50.10.140 11/27/06 Servlet.Engine.Transports : 1
size after: 4264 0000CBAx-81SRWWVmQfQ8-_47oK:-1
```

2. このフィーチャーは、HATS アプリケーションを WebSphere Portal Server で実行している場合には適用されません。
3. HATS でページを圧縮する前には、 HTTP ヘッダー "Accept-Encoding" に "gzip" が含まれていることを確認してください。
4. アプリケーションがサーバーで稼働中に指定した圧縮関連の設定を変更したい場合は、アプリケーション・サーバーを再始動する必要があります。
5. JavaScript (.js) ファイルおよび Cascading Stylesheet (.css) ファイルは、この新機能では圧縮されません。 HTTP サーバー内の HTTP 圧縮を、これらのファイル・タイプの圧縮に対応するように構成する必要があります。詳しくは、HTTP サーバーの資料を参照してください。
6. この設定は、プロジェクト・レベルでのみ指定できます。個々の変換 JSP には指定できません。
7. HATS 5.0.x、6.0、または 6.0.1 からプロジェクトをマイグレーションする場合は、該当するタグの付いた web.xml ファイルを手動で更新する必要があります。詳細については、 19 ページの『特別なマイグレーションの考慮事項』を参照してください。

未変更フィールドの送信を抑制する **Web のみ**

このボックスを選択すると、変更済みの入力フィールドの内容がホストが提供するデータと同じ場合には、HATS はその入力フィールドのデータをホストに送信しないように指定できます。

このボックスをクリアすると、変更済みの入力フィールドの内容がホストが提供するデータと同じ場合でも、HATS はその入力フィールドのデータを送信します。例えば、ホストがフィールドに ABC を設定していて、ユーザーがそのフィールドに ABC と入力した場合でも、入力されたデータはホストに返されます。

注: この設定は、プロジェクト・レベルでのみ指定できます。単一の変換 JSP には指定できません。

保護された読み取り専用ラベルをデフォルト・タブ順序に含める **RCP のみ**

このボックスを選択すると、「変換」ビューのフィールドのデフォルトのタブ移動順序に、保護された読み取り専用フィールドが含まれるように指定できます。

このチェック・ボックスをクリアすると、デフォルトのタブ移動順序には保護された読み取り専用フィールドは含まれません。

自動フィールド進行を使用可能にする

このボックスを選択すると、ユーザーが入力フィールドにデータを入れ終えたときに、フォーカスが自動的に次の入力フィールドに移るように指定できます。

Web アプリケーションでは、変換ファイルの </HATS:Form> タグの直後に以下の行を追加することで、個々の変換ごとにこの機能を使用不可にすることもできます。この機能を選択的に使用可能にするには、false の代わりに true を指定します。

```
<script>
autoAdvance = false;
</script>
```

リッチ・クライアント・アプリケーションでは、以下のように個々の変換で needsAutoAdvanceSupport メソッドをオーバーライドすることで、変換ごとにこの機能を使用不可にすることができます。この機能を使用可能にする場合は、false の代わりに true を戻します。

```
public boolean needsAutoAdvanceSupport(boolean preferredValue) {
return false;
}
```

注:

この設定を使用する場合の DBCS の考慮事項については、 546 ページの『自動フィールド進行を使用可能にする』を参照してください。

ホストおよび非ホスト入力フィールドを含める

このボックスを選択すると、自動拡張を有効にしたときに、ホスト画面の入力フィールドの順序に関係なく、変換された画面の入力フィールドの順で、フォーカスが自動的に次の入力フィールドに移るように指定できます。

このボックスをクリアすると、自動拡張を有効にしたときに、ホスト画面の入力フィールドの順で、フォーカスが自動的に次の入力フィールドに移ります。

初期カーソル位置

通常は、ホスト・アプリケーションのカーソルの位置によって、画面変換の初期入力フォーカスの位置が決まります。まれに、ホスト・アプリケーションのカーソル位置を無視し、代わりに変換での最初の項目に初期フォーカスを設定したい場合があります。高度にカスタマイズされた変換で、ホスト入力フィールドの順序が変更された場合がこれに該当します。このフィーチャーを使用可能にすると、初期フォーカスが、ホスト・カーソルがある入力フィールドではなく、画面変換の最初の項目に配置されます。

注:

1. 「**初期カーソル位置**」の設定は、 GUI では構成可能な設定として表示されません。変換ファイルのソースで設定する必要があります。

   Web アプリケーションの場合、変換でこのフィーチャーを使用可能にするには、変換の </HATS:Form> タグの直後に以下の行を追加します。

   ```
   <script>
   initialInputFocusFromCursor = false;
   </script>
   ```

リッチ・クライアント・アプリケーションの場合、変換でこのフィーチャーを使用可能にするには、以下のように個々の変換に対して isInitialFocusAtCursorPosition メソッドをオーバーライドします。

```
public boolean isInitialFocusAtCursorPosition() {
return false;
}
```

2. 変換のテストは慎重に行ってください。多くのホスト・アプリケーションでは、正しい振る舞いのためにはカーソル位置が重要です。例えば、変換に配置されたマクロ・ボタンが原因で、呼び出したマクロがホスト・アプリケーションの間違った位置にデータを入力し始めることがあります。これは、カーソルが想定されている位置とは別の位置にあるためです。

nextFieldForDropDown **Web のみ**

この設定を使用して、ドロップダウン・リストから選択した後で次の入力フィールドにカーソル位置を移動することを指定します。HATS V7.5.0.2 以降で作成される新規プロジェクトのデフォルトは、**true** です。HATS V7.5.0.2 より前のバージョンで作成されたプロジェクトのデフォルトは **false** です。

注:

1. GUI では、この設定は構成可能な設定として示されません。これは、次に示すように、application.hap ファイルのソースで設定する必要があります。

```
<class name="com.ibm.hats.common.RuntimeSettings">
        <setting name="nextFieldForDropDown" value="true"/>
</class>
```

2. この設定は、「**自動フィールド移動を有効にする**」が選択されている場合にのみ有効です。

上書きモード (初期)

初期状態で上書きモード (ブラウザーがサポートしている場合) を有効にする場合は、このボックスを選択します。上書きモードを使用可能にすると、入力フィールドにテキストを入力した場合、カーソル位置にあるテキストが一度に 1 文字ずつ上書きされます。これを使用不可にすると、入力フィールドに入力したテキストはカーソル位置に挿入され、既存のテキストは先送りされます。ユーザーは、Insert キーを使用して、この初期設定を切り替えることができます。

注:

この設定を使用する場合の DBCS の考慮事項については、546 ページの『上書きモード (初期)』を参照してください。

フォーカスされたテキストをすべて選択する

Web アプリケーションでよく見られるように、フィールドがフォーカスされた場合にそのフィールド内のテキストがすべて選択されるようにしたい場合は、このボックスを選択します。端末エミュレーターでよく見られるように、フィールドがフォーカスされた場合にテキストが選択されないようにしたい場合は、このボックスをクリアします。

注:

1. Web アプリケーションの場合:
   - デフォルトが選択されています。
   - この設定は、「**上書きモード (初期)**」設定の動作には影響しません。
   - この設定は、アプリケーションのブラウザーとして Internet Explorer が使用される場合に限り有効です。
2. リッチ・クライアント アプリケーションの場合:
   - デフォルトではクリアされています。
   - 選択すると、この設定は「**上書きモード (初期)**」設定のように機能します。ユーザーがフィールドに入力を行うと文字が上書きされていきます。
   - キーボードを使用してフィールドに Tab で移動した場合に限り、テキストが選択されます。フィールド内でマウスをクリックしても、テキストは選択されません。
3. この設定を使用する場合の DBCS の考慮事項については、546 ページの『フォーカスされたテキストをすべて選択する』を参照してください。

使用中のページを有効にする **Web のみ**

このボックスを選択すると、最初の要求の処理が完了する前にユーザーによって複数の要求がサブミットされたときに、使用中ページのメッセージが表示されます。使用中ページのメッセージを表示させない場合は、このボックスをクリアします。クリアされている場合は、サーバーが応答を戻すまでは追加の要求を送信できません。

矢印キー・ナビゲーションを使用可能にする **RCP のみ**

レンダリングされた画面上の無保護ホスト・フィールド間を上下左右の矢印キーを使用してナビゲートできるようにするには、このボックスを選択します。「レンダリング」タブのフィールド・ウィジェットで、「**保護フィールド上でのカーソルの位置決めを許可します**」設定も選択されている場合は、矢印キーを使用して、無保護と保護の両方のホスト・フィールドにナビゲートすることができます。これは、端末エミュレーター・アプリケーションの標準動作です。矢印キーをナビゲーションに使用しない場合は、このボックスをクリアします。

各変換の needsArrowKeyNavigationSupport メソッドを以下のようにオーバーライドすることにより、この機能を個々の変換について選択的に無効化することができます。

```
public boolean needsArrowKeyNavigationSupport(boolean preferredValue) {
return false;
}
```

この機能を選択的に使用可能にするには、`false` の代わりに `true` を戻します。

注:

1. ナビゲーションは、フィールドの UI 位置ではなく、常にホスト・フィールドの位置が基準になります。
2. 複数の画面と組み合わせてこの機能を使用した場合、次の入力フィールドを判断するアルゴリズムは、まず、現在のフィールドと同じ画面にあ

る対象フィールドへとナビゲートしようとします。対象フィールドがない場合には、次の画面 (画面の ID を基準に判断) に対象となる入力フィールドがないか探します。

先行入力サポートを使用可能にする RCP のみ

先行入力サポートを使用可能にするには、このボックスを選択します。このサポートが使用可能になると、ユーザーはホストから送信される次の画面が HATS で受信されて処理される前に、その画面の入力フィールドを対象とするデータを入力し始めることができます。次の画面が受信されると、HATS は、入力をホストに送信するキーを含め、既に入力されていたデータ (先行入力データ) を送信します。

これらの画面に画面カスタマイズがある場合、変換の適用を含め、カスタマイズで定義されているすべてのアクションが実行されます。ただし、先行入力データは、ホストに送信されるまで、実際にはレンダリングされた変換に挿入されません。その代わり、先行入力データは以前に入力されたとおりにホストに直接送信され、変換は完全にバイパスされます。このように、このサポートは端末エミュレーターの先行入力サポートのしくみと非常によく似ています。このことはユーザーにとって、先行入力データは、適用された変換が予期する内容ではなく、それぞれのネイティブ・ホスト画面が予期する内容に適合している必要があるということを意味します。

このボックスをクリアすると、「**先行入力フィールド**」設定が、「**レンダリング**」タブの「**オペレーター情報域**」でクリアされます。オペレーター情報域の先行入力データの表示について詳しくは、123 ページの『先行入力フィールド』を参照してください。

注:

1. ホスト補助キーとそのホスト・データ (例えば、`userid[tab]password[enter]`)、およびホスト・ナビゲーション・キー (例えば、**tab** および **home**) のみ先行入力可能です。
2. アプリケーション・キーとコマンド (例えば、「切断」コマンド。これは、キー・シーケンス **CTRL+D** のデフォルト・コマンドです) は先行入力できません。
3. RCP ランタイム拡張プラグイン・レベルのキー再マップが、先行入力モードのホスト・キーとコマンドでサポートされています。

マクロ・コンテンツ・アシスタンス

コンテンツ・アシスタンスで使用可能にするマクロを選択するには、以下の設定を使用します。 Macro Editor または Visual Macro Editor の「**ソース**」タブでマクロを編集中にコンテンツ・アシスタンスを起動するには、Ctrl+Space を押してください。

すべてのマクロ

プロジェクト内のすべてのマクロでコンテンツ・アシスタンスを使用可能にするには、このラジオ・ボタンを選択します。

選択したマクロ (Selected macros)

マクロのリストで選択したマクロでのみコンテンツ・アシスタンスを使用可能にするには、このラジオ・ボタンを選択します。

マクロ コンテンツ・アシスタンスで使用可能にするマクロを選択します。

全選択 「**選択したマクロ (Selected macros)**」ラジオ・ボタンを選択した場合にのみ使用可能になります。リスト内のすべてのマクロを選択するには、このボタンをクリックします。

全選択解除

「**選択したマクロ (Selected macros)**」ラジオ・ボタンを選択した場合にのみ使用可能になります。リスト内のすべてのマクロの選択をクリアするには、このボタンをクリックします。

ポートレットの設定

これらの設定は、HATS 標準 (JSR 168 または JSR 286) ポートレット・プロジェクトにのみ適用されます。これらは、ポートレット通信ソリューションを開発する手助けとして使用できます。詳しくは、462 ページの『ポートレット通信』を参照してください。双方向言語サポートを使用する場合の考慮事項については、531 ページの『ポートレット・サポート』を参照してください。

JSR 168 ポートレット

これらの設定を使用して、ポータル・サーバー上の他の JSR 168 ポートレット (HATS または非 HATS) から WebSphere Portal を介して受信するプロパティーを定義します。新規プロパティーを追加するには、「**追加**」をクリックします。「プロパティー」フィールドに、受信するプロパティーの名前を入力します。名前は、送信ポートレットで使用されるプロパティー名に一致する必要があります。例えば、HATS JSR 168 送信ポートレットの「**グローバル変数の送信**」アクションで使用されているプロパティー名と同じにします。「グローバル変数」フィールドに、受信したプロパティーの値を格納するために使用するグローバル変数の名前を入力します。グローバル変数が索引付きである場合は、使用する索引を入力します。グローバル変数が共用グローバル変数である場合は、「**共用**」チェック・ボックスを選択します。

「**完了**」をクリックすると、プロジェクトの application.hap ファイルにプロパティーが追加されます。また、Web サービス記述言語 (WSDL) ファイルがまだない場合は作成され、その WSDL ファイルに受信プロパティーが定義されます。 WSDL ファイルの名前は <project_name>.wsdl です。 WSDL ファイルは、WebSphere Portal ワイヤリング・ツールを使用してポートレットをワイヤリングするために必要です。 portlet.xml ファイルは、WSDL ファイルの場所を参照するように更新されます。

「**除去**」をクリックした場合、プロジェクトの application.hap ファイルからはプロパティーが除去されますが、WSDL ファイルからは除去されません。 WSDL ファイルを編集し、WSDL エディターを使用してプロパティーを除去できます。

注: WSDL ファイルからプロパティーを除去しなければ、HATS ポートレットは現状のまま正常に稼働します。

JSR 286 ポートレット

これらの設定を使用して、ポータル・サーバー上の他の JSR 286 ポートレット (HATS または非 HATS) から WebSphere Portal を介して受信するイベントを定義します。新規イベントを追加するには、「**追加**」をクリックします。「イベント名」フィールドに、受信するイベントの名前を入力します。 名前は、送信ポートレ

ットで使用されるイベント名に一致する必要があります。例えば、HATS JSR 286 送信ポートレットの「**グローバル変数の送信**」アクションで使用されているイベント名と同じにします。 「グローバル変数」フィールドに、受信したイベントの値を格納するために使用するグローバル変数の名前を入力します。

注: 単一ポートレット内のイベント名は固有でなければなりません。

必要な場合、「**拡張**」をクリックしてさらにオプションを設定します。

「詳細オプション」ページで、「**グローバル変数を新しい値で上書き**」、または「**グローバル変数の特定の索引を新しい値で上書き (Overwrite a specific index of the global variable with the new value)**」のいずれかを指定し、上書きする「**索引**」およびオブジェクトの「**イベント・ペイロード・タイプ**」を含めます。「**参照**」ボタンを使用して、有効な値タイプを入力します。選択するタイプは、順序付け可能である必要があります。順序付け可能でない場合、エラー・メッセージが表示されます。タイプも、送信ポートレットで使用されるタイプに一致する必要があります。例えば、HATS JSR 286 送信ポートレットの「**グローバル変数の送信**」アクションで使用されているタイプと同じにします。 グローバル変数が共用グローバル変数である場合は、「**共用**」チェック・ボックスを選択します。

注: 「**イベント・ペイロード・タイプ**」指定は、JSR 286 ポートレットの場合にのみサポートされます。JSR 168 プロジェクトの application.hap ファイルのソースにタイプ・パラメーターを手動で追加すると、それは無視されます。

「**終了**」をクリックすると、イベントがプロジェクトの application.hap ファイルに追加され、portlet.xml ファイルが更新されます。JSR 286 ポートレット通信に関与する WSDL ファイルはありません。

ソース

「**ソース**」タブには、プロジェクト内のすべての設定 (自分で選択したもの、またはデフォルトを受け入れたもの) について、application.hap ファイルのタグと値が表示されます。プロジェクト・エディターの他のタブを変更すると、「**ソース**」タブに表示されるタグと値もそれに合わせて変化します。

「**ソース**」タブのソースを編集することにより、application.hap ファイルのタグと値を直接変更することもできます。この変更は、プロジェクト・エディターの適切なタブに反映されます。

HATS プリファレンスの使用

「HATS プリファレンス」を使用して、HATS Toolkit の外観および動作を制御できます。これらのプリファレンスによって、HATS Toolkit へのアクセスが容易になり、プログラミング・タスクが単純化されます。HATS プリファレンスを処理するには、「**ウィンドウ**」>「**設定**」をクリックして、ツリー・ビューで「**HATS**」を選択します。プリファレンスに関する即時ヘルプを参照するには、その場所にカーソルを置いて **F1** を押します。

以下のプリファレンスを変更できます。

既知のファイル・タイプのファイル拡張子は非表示

デフォルトでは、「**HATS Projects**」ビューに .jsp などの一般的なファイル拡張子は表示されません。例えば、**Templates** フォルダーのファイルの拡張子はすべて .jsp です。「**HATS Projects**」ビューにすべてのファイル拡張子を表示させる場合は、このチェック・ボックスをクリアします。

HATS のヒントを表示

HATS のヒントは、新規ユーザーに役立つ提案を表示します。HATS Toolkit を使用するときに HATS のヒントを表示させないようにする場合は、このチェック・ボックスをクリアします。

マイグレーション警告の表示

HATS V9.5 プロジェクトにマイグレーションされていないプロジェクトを HATS が検出した場合に、マイグレーションの警告を表示させるかどうかを指定できます。

端末ウィンドウを表示するためのプロンプト

「サーバーでデバッグ」(Web プロジェクトの場合) または「デバッグ」(リッチ・クライアント・プロジェクトの場合) を使用してプロジェクトをテストする際に、ディスプレイ端末をオンにするか尋ねるダイアログを、HATS で表示するかどうかを決定できます。ダイアログ上のチェック・ボックスをポップアップ表示されたときに選択することによって、この機能をオフにすることもできます。

隠しフィールド抽出の警告を表示 (Show extract hidden fields warning)

以下のイベントのいずれかが発生した場合、キャプチャーされた画面で HATS が隠しフィールドを検出した場合にダイアログによる警告を表示するかどうかを指定できます。

- ライブ端末によるマクロの記録中にデータを抽出している場合
- キャプチャーされた画面を使用して画面カスタマイズ基準を作成している場合
- キャプチャーされた画面を使用して新規の「グローバル変数を抽出」アクションを追加している場合

リッチ・クライアント成果物の完全修飾 Java クラス名を表示する

このボックスを選択すると、HATS Java ソース成果物 (例えば、リッチ・クライアント・テンプレートやリッチ・クライアント変換) の完全修飾パッケージ名が、「**HATS プロジェクト**」ビューに表示されます。このボックスはデフォルトでは選択されていないので、「**HATS プロジェクト**」ビューには成果物の Java クラス名のみが表示されます。例えば、リッチ・クライアント・プロジェクトに `myProj.transformations.SignOn` という完全修飾名の変換が含まれているとします。デフォルトでは、「**HATS プロジェクト**」ビューの「**変換**」フォルダーには `SignOn` しか表示されません。このボックスが選択されていると、`SignOn (myProj.transformations)` と表示されます。

この設定は、**Java**「**HATS プロジェクト**」ビューの「**Java ソース**」フォルダーの表示内容には影響を与えません。

JRE の非互換性に関する警告を表示

HATS では、非互換の JRE を検出したとき、メッセージにより警告を出すことができます。現行 JRE が HATS と非互換であるという警告ダイアロ

グを表示する場合は、このボックスを選択してください。 JRE を手動で変更する必要があります。警告ダイアログを表示しない場合は、このボックスをクリアしてください。

ワークベンチの始動時にライセンス設定検査を実行する

HATS はワークスペースの始動時にプロジェクトのライセンス設定をチェックし、マスター・ライセンス設定と一致しないプロジェクトがある場合は、ユーザーにライセンス設定の更新を求めるプロンプトを表示します。このチェック・ボックスをクリアすると、始動時にライセンス設定を確認することはできません。

選択フィルター

このチェック・ボックスのリストにより、「**HATS Projects**」ビューに表示させるファイルのタイプを選択できます。例えば、共通イメージやスタイル・シートを処理しない場合は、このチェック・ボックスをクリアして「**HATS Projects**」ビューに表示されないようにすることができます。経験の少ないユーザーにマクロや Java ソース・ファイルを処理させたくない場合、該当するチェック・ボックスをクリアして、それらのファイルが表示されないようにすることができます。

ホスト端末プリファレンス

マクロのデバッグ前にすべてのマクロ・プロンプトを表示 (Prompt for all macro prompts before debugging a macro)

これを選択した場合、マクロを最初にデバッグするときにダイアログが表示され、すべてのプロンプトに対する値を一度に入力できます。選択しない場合、マクロが実行している現在のプロンプトの値のみを要求するダイアログがプロンプト表示されます。この場合、各プロンプトの値を 1 つずつ入力する必要があります。このプリファレンスは、マクロのデバッグ時だけに使用され、HATS Toolkit でマクロを再生する場合、またはマクロや HATS アプリケーションを実行する場合には影響ありません。

マクロ・プロンプト/抽出名の警告を表示

このボックスは、統合オブジェクトとの使用には無効なマクロ・プロンプトまたは抽出名を入力したときに HATS に警告させる場合に選択します。

固定フォント・サイズ

このプリファレンスは、視覚障害のあるユーザーのために設計されています。ウィンドウを拡張したり縮小したりした場合でもホスト端末のテキストのサイズを変更しないように指定できます。例えば、文字を大きくする必要がある場合は、大きな固定フォント・サイズを設定し、スクロール・バーを使用してホスト端末画面を参照できます。

画面キャプチャーの強調表示時の色

色を使用して、パターンに一致する入力フィールド、保護フィールド、隠しフィールド、および画面領域を識別します。これらの目的に使用する色を変更するには、これらのプリファレンスを使用します。

HATS に関連した動作を制御するためのプリファレンスはほかにもあります。これらのプリファレンスを使用するには、プリファレンス・ツリーで **HATS** を展開し、以下のサブカテゴリーの 1 つをクリックします。

アクセシビリティー

音響アラームによる認識エラーのアラート
認識エラーが発生した場合 (例えば、「サブファイル」コンポーネント設定ページなどで) に音響アラート (ビープ音) を再生するには、このボックスを選択します。これは、スクリーン・リーダー (読み上げソフトウェア) を使用してメッセージ領域を読み上げるよう、開発者に警告しなければならない場合に便利です。

BMS のインポート (BMS Import)
これらの設定は、BMS インポート・ウィザードを使用せずに、BMS ソース・ファイルを HATS プロジェクト内の **Maps** フォルダーに追加した場合だけに使用されます。

ホスト・コード・ページ
インポートする BMS マップで使用されるコード・ページ。

BMS ファイル・コード・ページ
BMS ファイルをワークステーションに転送するときに使用されるコード・ページ

デフォルト Java パッケージ
このパネルで、Java ソース・ファイルである HATS 成果物 (リッチ・クライアント変換、リッチ・クライアント・テンプレート、リッチ・クライアント・マクロ・ハンドラー、ビジネス・ロジック、カスタム・ウィジェット、カスタム・コンポーネント、および RESTful サービス) のデフォルト Java パッケージ名を構成します。パッケージ名は、完全修飾 Java クラス名の最初の部分です (例えば、Java クラス名が `com.myCompany.MyClass` であれば、パッケージ名は `com.myCompany` です)。

新規成果物にパッケージ名を作成する場合、 `{project name}` にはプロジェクトの名前が入ります。このような置換が行われるのは、有効なパッケージ名を確実に作成するためです (例えば、スペースは下線に変換されます)。

ホスト・シミュレーション

TCP/IP ポートの範囲
トレースの記録および再生中に、HATS ホスト・シミュレーション機能はポートの範囲を使用して、ホスト端末または HATS アプリケーションとのホスト通信を記録または再生します。ポートの範囲を使用すると、複数のセッションで同時にトレースを記録または再生することができます。ポートの範囲を設定するには、次のフィールドを使用します。デフォルトの範囲開始ポートは **7021**、デフォルトの終了ポートは **7050** です。

時間遅延設定

遅延 このドロップダウンの各オプションは、ホスト・シミュレーションが、再生中にホスト端末または HATS アプリケーションからの要求応答するまでの時間遅延を設定するために使用します。オプションは、「**遅延なし**」(ホスト・シミュレ

ーションは次の画面ですぐに応答します)、「**ランダム**」(ホスト・シミュレーションが応答するまでに、あらかじめ定義された最小時間と最大時間の間でランダムに遅延が発生します)、および「**実**」(ホスト・シミュレーションは、記録中に画面を表示する場合に実際にかかる時間だけ待機します) です。デフォルトは「**遅延なし**」です。「**ランダム**」を選択した場合は、「**最小**」フィールドと「**最大**」フィールドで遅延値の範囲を設定します。

最小 (ms)
「**遅延**」設定が「**ランダム**」の場合は、このフィールドで、ホスト・シミュレーションが再生中に次の画面で応答するまでの遅延の最小値 (ミリ秒) を設定します。

最大
「**遅延**」設定が「**ランダム**」の場合は、このフィールドで、ホスト・シミュレーションが再生中に次の画面で応答するまでの遅延の最大値 (ミリ秒) を設定します。

統合オブジェクト (Integration Object)

マクロ保存時に統合オブジェクトを自動的に生成
マクロを保存するたびに統合オブジェクトを生成する場合は、このチェック・ボックスを選択します。手動で統合オブジェクトを作成する場合は、このチェック・ボックスをクリアします。マクロから統合オブジェクトを作成する方法については、403 ページの『第 13 章 統合オブジェクトの使用』を参照してください。

統合オブジェクトの作成時に HATS EJB プロジェクトで EJB Access Bean を自動生成する (Automatically generate EJB Access Bean in HATS EJB Project when an Integration Object is created)
統合オブジェクトを作成するたびに HATS EJB プロジェクトに EJB Access Bean を生成する場合は、このチェック・ボックスを選択します。「**マクロ保存時に統合オブジェクトを自動的に生成**」プリファレンスを選択した場合、マクロの作成時または変更時に、統合オブジェクトとともに EJB Access Bean が作成されます。

統合オブジェクトの生成に使用するテンプレート
統合オブジェクトの新規テンプレートを作成する場合は「*HATS Web アプリケーション・プログラマーズ・ガイド*」で説明されているとおりに、この設定を使用して、統合オブジェクト・ファイルを作成する際に使用するテンプレートを指定します。

統合オブジェクト BeanInfo の生成に使用するテンプレート
統合オブジェクトの新規テンプレートを作成する場合は「*HATS Web アプリケーション・プログラマーズ・ガイド*」で説明されているとおりに、この設定を使用して、統合オブジェクトの BeanInfo ファイルの定義時に使用するテンプレートを指定します。

ターゲット・プラットフォーム RCP のみ

HATS ランタイム・プラグインをターゲット・プラットフォームにインストールするためのプロンプト (リッチ・クライアントのみ)
HATS リッチ・クライアント・プラグインをターゲット・プラットフォームのローカル・テスト環境でテストするためには、HATS ラ

ンタイム・プラグインをターゲット・プラットフォームにインストールする必要があります。新規プロジェクトを作成してターゲット・プラットフォームを選択する場合、またはあとでターゲット・プラットフォームを変更する場合、HATS Toolkit は、選択されているターゲット・プラットフォームに HATS ランタイム・プラグインがインストールされているかどうかを検出します。このボックスを選択すると、ターゲット・プラットフォームでランタイム・プラグインが検出されない場合、ランタイム・プラグインをインストールするように求めるプロンプトがツールキットから出されます。

変換 これらのプリファレンスは、新規のブランク変換を作成する際に使用されます。

フリー・レイアウト・テーブルを含める (Web のみ)
ブランクの変換を作成する場合、テーブルのデフォルトの高さと幅がこの設定から取得されます。デフォルトでは、特に別の指定をしない限り、この設定が 100% で初期化されます。ドロップダウン・メニューから「**ピクセル**」を選択すると、これらの値をピクセルで指定することもできます。

注: 変換でテーブルの後ろにホスト・コンポーネントが挿入されると、サーバーで実行したときに、レンダリングされたそのコンポーネントは表示されません。これは、画面の 100% がテーブルに使用されるためです。

プロジェクトがモバイル装置用に最適化されている場合を除く
モバイル装置用に最適化されているプロジェクトの新規ブランク変換にフリー・レイアウト・テーブルを組み込まないようにするには、このプリファレンスを選択します。

ホスト・コンポーネントの挿入後にプロパティー・ビューをオープン
変換へのホスト・コンポーネントの挿入後に「プロパティー」ビューをオープンするには、このボックスを選択します。

Visual Macro Editor

デフォルトで画面アクションを表示
この設定を選択すると、デフォルトで、マクロ画面オブジェクト上にアクションを表示します。この設定をクリアすると、デフォルトで、アクションを非表示にします。マクロ画面上のアクションをトグルでクリックすると、特定の画面のアクションを表示/非表示にします。この設定は、Visual Macro Editor で初めて開くマクロのみに作用します。デフォルトが選択されています。

表示する画面アクション数の制限
1 つのマクロ画面オブジェクトに対して表示されるアクションの最大数を指定します。これは、すべてのアクションを表示するとインターフェースが複雑すぎる、複雑なマクロの場合、有用です。デフォルトは「**5**」です。

HATS プリファレンスを変更した場合、「**適用**」をクリックして即時 HATS Toolkit にその結果を表示することができます。新しいプリファレンスを保存して「プリファレンス」ウィンドウをクローズするには「**OK**」をクリックします。保存せずに終

了するには「**キャンセル**」をクリックします。「**キャンセル**」 をクリックした場合は、「**適用**」をクリックした後に行った変更内容のみがキャンセルされます。

HATS プリファレンスを初期値に戻すには、「**デフォルトの復元 (Restore defaults)**」をクリックします。

ある HATS Toolkit で HATS プリファレンスを構成し、そのプリファレンスをほかの開発ワークステーションにコピーする場合、Rational SDP メニュー・バーから「**ファイル**」>「**エクスポート**」>「**一般**」>「**設定**」を選択して、プリファレンスを .epf ファイルとして保存します。このファイルを使用する各ワークステーションに転送したり、ネットワーク経由で使用可能にすることができます。「**ファイル**」>「**インポート**」>「**一般**」>「**設定**」を選択し、設定ファイルをインポートします。

その他の Rational SDP プリファレンスの使用

HATS Toolkit の外観と振る舞いは、Rational SDP で選択できる他のプリファレンスを使用することによって影響を受ける場合もあります。

例えば、「ラベル装飾」プリファレンスを使用すると、他のツールキット (Java コンパイルおよび CVS など) で、「HATS Projects」ビューに表示される HATS ラベルおよびイメージを装飾することができます。エラーが発生した場合に、ファイルの横に赤いマークを表示することができます。CVS リポジトリーからファイルをチェックアウトする場合に、ファイル名の横にバージョン番号を表示できます。Rational Application Developer で「ラベル装飾」プリファレンスにアクセスする手順は次のとおりです。

1. メニュー・バーから、「**ウィンドウ**」>「**設定**」を選択します。
2. 「**一般**」>「**外観**」の順に、ツリー・ビューを展開します。
3. 「**ラベル装飾**」を選択します。

注:

a. 装飾のプリファレンスは、Rational SDP の他のバージョンでも同じ場所にあります。

b. 詳しくは、Rational SDP のヘルプ・システムで「ラベル装飾」を検索してください。

第 6 章 接続の管理

接続は、.hco ファイルに保管された一連のパラメーターであり、 HATS がホスト・アプリケーションに接続する際に使用されます。HATS には、デフォルト接続 (変換接続とも呼ばれる) とバックグラウンド接続の 2 つの接続のタイプがあります。各 HATS アプリケーションでは、HATS によって変換される画面を持つホスト・アプリケーション用のデフォルト接続を 1 つ保持しています。バックグラウンド接続は、HATS アプリケーションにおけるデフォルト接続以外の任意の接続で、統合オブジェクトまたは「マクロ・トランザクションを実行」画面イベント・アクションで使用できます。HATS では、バックグラウンド接続からの画面の変換は実行されません。ただし、デフォルト接続として使用する接続を動的に選択できます。詳しくは、 131 ページの『パラメーターのオーバーライド時に使用する接続のタイプの選択』を参照してください。

バックグラウンド接続は、デフォルト接続と同じホストに対する別の接続インスタンスにすることも、完全に別のバックエンド・ホストへの接続にすることもできます。バックグラウンド接続によって、別のホストと対話することができます。事前記録したマクロを使用して、データの収集、入力、またはホスト間でのデータ交換を行うことができます。このデータは、デフォルト接続とも組み合わせることができます。このようにして、ほかのホスト上のトランザクションを自動化できるだけでなく、ユーザーのデフォルト接続を変換することもできます。バックグラウンド接続はプールすることができます。

VT ホストは、バックグラウンド接続に制限されます。バックグラウンド接続の 1 つとして使用する VT 接続は、マクロの記録と統合オブジェクトの実行に使用することができます。詳しくは、 193 ページの『マクロ・トランザクションを実行』を参照してください。

接続の作成

「接続の作成」ウィザードを使用すると、より多くの接続構成を作成することができます。このウィザードを使用するには、「HATS Projects」ビューのポップアップ・メニューを使用することができます。メニュー・バーの「**HATS**」>「**新規**」>「**接続**」または「**ファイル**」>「**新規**」>「**HATS 接続**」を選択するか、またはツールバーの「**HATS 接続の作成**」アイコンをクリックします。

「接続の作成」ウィザードが表示されます。このウィザードでは、ドロップダウン・リストからターゲット・プロジェクトを選択し、接続に名前を指定し、説明を入力し、接続定義の保存先を確認できます。これらの項目を指定したら、「**次へ**」をクリックします。

次に、「**接続設定**」ページが表示されます。このウィザードでは、ホスト名、ポート、端末タイプ、コード・ページ、および画面サイズなどの基本的な接続情報が必要になります。これらの設定について詳しくは、 158 ページの『基本』を参照してください。

© Copyright IBM Corp. 2003, 2015

このウィザードを使用して、一部の拡張接続設定を指定することもできます。表示される拡張接続設定は、接続タイプによって異なります。接続が 5250 の場合は、ワークステーション ID を構成できます。3270E 接続を使用する場合は、LU またはプール名を構成できます。5250W、3270、または VT 接続ではワークステーション ID または LU 設定はサポートされていません。 これらの設定について詳しくは、159 ページの『拡張』を参照してください。

接続エディター

接続エディターを使用すると、接続をカスタム構成できます。また、このエディターは、「概要」、「基本」、「拡張」、「印刷」、「画面処理」、「セキュリティー」、「プール」、「マクロ」、「ユーザー・リスト」、および「ソース」の各タブを持ちます。

概要

接続エディターの「**概要**」タブには、接続の作成時にユーザーが指定した接続設定がほぼ要約されています。このタブ上で変更できるのは、接続の説明のみです。

「**概要**」タブ上のセクション見出しは、それぞれ、接続エディターの残りのタブへのリンクになっています。

基本

「**基本**」タブに、「接続の作成」ウィザードで最初にセットアップした設定を含む接続の基本設定が表示されます。以下の設定を変更できます。

ホスト名
接続が接続されているホストの名前。これは、Telnet サーバーの名前または IP アドレスです。5250W 接続 (以下の「**タイプ**」フィールドを参照) の場合は、これは WebFacing サーバーの名前または IP アドレスです。

Host On-Demand バージョン 8 では、インターネット・プロトコル・バージョン (IPv6) がサポートされています。WebSphere Application Server バージョン 6 でも IPv6 がサポートされます。HATS では、IPv6 がサポートされるプラットフォームで IPv6 の機能を利用できるように、IPv6 アドレスを構成できます。

注: HATS は、Linux、Solaris、および AIX® 上でのみ IPv6 をサポートします。IPv6 アドレスを使用する Windows HATS Toolkit 環境には制限があります。そのため、このタイプのアドレスを使用する場合、ホスト端末機能を使用したり、ローカル・テスト・サーバーをテストしたりできないことがあります。

タイプ 接続用のホスト・セッションのタイプ。3270、3270E、5250、5250W、VT ホスト (VT52、VT100、VT420_7、および VT420_8) などです。

注:

1. HATS では、変換接続 (デフォルト) の接続タイプとして 3270、3270E、5250、および 5250W のみがサポートされています。
2. バックグラウンド接続は、どのタイプでもかまいません。

3. 5250W 接続の詳細については、 473 ページの『WebFacing サーバーへの HATS 接続』を参照してください。
4. VT220 および VT320 は VT420 のサブセットです。HATS で VT220 ホストまたは VT320 ホストをサポートするには、VT420_7 (ご使用の VT ホストが 7 ビット・コマンドを送信する場合) または VT420_8 (ご使用の VT ホストが 8 ビット・コマンドを送信する場合) を指定します。

ポート 接続で経由するポート番号。一般に、ポート 23 は Telnet サーバー接続に使用され、ポート 4004 は WebFacing サーバー接続に使用されます。これとは異なるポート番号を使用している場合は、ここでそれを指定する必要があります。

コード・ページ
接続に使用するコード・ページ。バックエンド・ホストのコード・ページを選択します。

注:

1. 双方向 (BIDI) コード・ページを選択する場合は、 525 ページの『ユーザーによる画面方向反転の可能化』を参照してください。
2. アラビア語コード・ページ 420 を選択した場合、 535 ページの『アラビア語の選択的形状』および 536 ページの『アラビア (西洋) 数字の入力を無効にする』を参照してください。
3. 5250 接続に対し中国語 (繁体字) コード・ページ 937 を選択した場合に、アクセント付き文字を使用できるようにするには、 509 ページの『コード・ページ 937 用のアクセント付き文字の使用』を参照してください。

画面サイズ
ホスト接続の画面サイズ。この画面サイズによって、ホスト画面の行数と列数が決まります。

注: 画面サイズに有効な値は、選択するコード・ページおよびホスト・セッション・タイプによって異なります。

ライブ接続の代わりに、ホスト・シミュレーションを使用
この接続の接続アクション (「**ホスト端末をオープン**」、「**実行**」、「**サーバーで実行**」 など) で、ライブ接続の代わりにホスト・シミュレーション・トレース再生を使用することを指定するには、このボックスを選択します。このボックスを選択した場合は、ドロップダウン・リストから使用するトレース・ファイルも選択してください。詳細については、 437 ページの『第 17 章 ホスト・シミュレーションの使用』を参照してください。

注: このオプションは、プロジェクトにホスト・シミュレーション・トレースが存在している場合にのみ選択できます。

拡張

「**拡張**」タブに、「接続の作成」ウィザードで最初にセットアップした設定を除いた接続の拡張設定が表示されます。ホスト、セッション・タイプ、ポート、コード・ページ、および画面サイズなどの基本設定を変更するには、「基本」タブを使用してください。 以下の拡張設定は、「拡張」タブで変更することができます。

3270E 接続の場合は、接続に LU 名または LU プール名を指定できます。 LU 名または LU プール名を指定するための 4 つのオプションがあります。次のオプションのいずれかを選択します。

なし (サーバーで割り当て)
Telnet サーバーで LU を割り当てる場合は、これをクリックします。このオプションを使用できるのは、プールだけです。

プロンプトを出す
LU 名を指定するようにユーザーに対してプロンプトを表示する場合は、これをクリックします。プールを使用可能にしている場合、このオプションを使用しないでください。

指定値を使用
使用される LU 名または LU プール名を指定するには、これを使用します。接続プールを使用するつもりの場合、定義する接続プールをサポートするのに十分な大きさの LU プールの名前を指定します。

HTTP セッション変数を使用 **Web のみ**
LU 名または LU プール名として HTTP セッション変数の値を指定する場合は、これをクリックします。プールを使用可能にしている場合、このオプションを使用しないでください。

5250 接続の場合は、接続にワークステーション ID を指定できます。ワークステーション ID を指定するための 4 つのオプションがあります。次のオプションのいずれかを選択します。

なし (サーバーで割り当て)
Telnet サーバーでワークステーション ID を割り当てる場合は、これをクリックします。このオプションを使用できるのは、プールだけです。

プロンプトを出す
ワークステーション ID を指定するようにユーザーに対してプロンプトを表示する場合は、これをクリックします。プールを使用可能にしている場合、このオプションを使用しないでください。

指定値を使用
ワークステーション ID の作成元となったストリングを指定する場合は、これをクリックします。このオプションは、ワークステーション ID をワイルドカードとして指定した場合に、プールだけで使用できます。

5250 接続のワークステーション ID パラメーターには、ワイルドカード文字を使用できます。ワークステーション ID では、ワークステーションの名前を定義します。最初の文字は、A から Z、$ (ドル記号)、@ (単価記号)、または # (番号記号) のいずれかである必要があります。その他の文字には、A から Z、0 から 9、$、@、#、. (ピリオド)、および _ (アンダースコアー) のいずれかを指定できます。このフィールドに入力しない場合は、ホストによって自動的にワークステーション ID が割り当てられます。HATS サーバーによって、セッションに対する特定のワークステーション ID を新規作成することができます。ワークステーション ID フィールド内にキーワードと特殊文字を指定すると、次の情報の一部または全部が、Telnet サーバーに送信されるワークステーション ID の値として代入されます。

- 短縮セッション ID(*)
- セッション・タイプ ID(%)
- 衝突回避 ID(=)

衝突回避 ID を指定すると、Telnet サーバーで以前の名前が拒否された場合に (以前の名前が IBM i サーバーで既に使用されている場合に発生)、新規ワークステーション ID が作成されます。

ワークステーション ID フィールドでワイルドカードを組み合わせて指定すると、ホストからの接続受け入れを獲得するために、HATS サーバーによってより多くのバリエーションの ID が自動的に生成されます。その結果、ホストからクライアントに対して有効なワークステーション ID の送信要求が再発行される回数が減るので、セッション接続に必要な時間が短縮されます。

ワイルドカード文字は、ワークステーション ID フィールド内に他の英数字と組み合わせて複数指定してもかまいません。(N=A=M=E、NAME== など。) 各ワイルドカードを使用すればするほど、すでに他のセッションで要求されたワークステーション ID が生成される可能性が低くなります。

- 「=」を 2 文字使用するだけで、確立済みの ID と重複する可能性が約 1/50 になります。
- 「=」を 3 文字使用すると、確立済みの ID と重複する可能性が約 1/300 になります。
- 生成される固有の名前の数は、約 36^n です。この *n* は、等号 (=) の数です。

例えば、ワークステーション ID として NA=ME を指定すると、「=」によって、英数字「N、A、M、E」を使用した 36 個の固有の ID を生成できます。「=」をもう 1 つ追加すると、約 1296 の固有 ID の生成が起動されるなどです。

HTTP セッション変数を使用 **Web のみ**

ワークステーション ID として HTTP セッション変数の値を指定する場合は、これをクリックします。プールを使用可能にしている場合、このオプションを使用しないでください。指定するストリングにワイルドカード文字を使用することもできます。詳しくは、 160 ページの指定値を使用を参照してください。

「オプションの拡張接続設定を構成してください」セクションでは、パラメーター・テーブルの右側のボタンを使用して、追加 IBM Host On-Demand セッション・パラメーターを追加、変更、または削除できます。**「追加」**をクリックすると、**「名前」**フィールドの横のドロップダウン・リストを使用してパラメーターを選択してから、値フィールドに値を入力できます。一部のパラメーターは、接続エディターの他のタブで設定されます。それらをここで設定しようとすると、警告メッセージが表示されます。詳しくは、Host On-Demand Knowledge Center (http://www-01.ibm.com/support/knowledgecenter/SSS9FA_11.0.0/com.ibm.hod.doc/WebSphereHOD.htm) にアクセスして、セッション・パラメーターに関して検索を行ってください。

注:

1. 「基本」タブおよび「拡張」タブにフィールドを持つパラメーターは、このテーブルに入力できません。これらのパラメーターは、そのパラメーター固有のフィールドで変更する必要があります。
2. ここで追加される印刷サポート・パラメーターは無視されます。印刷サポート・パラメーターは「印刷」タブで追加してください。詳しくは、『印刷』を参照してください。
3. 3270E 接続の場合は、デフォルトで **negotiateCResolution** パラメーターが **true** に設定されています。詳細情報と、この設定を **false** に変更する状況については、573 ページの『z/OS Communications Server を使用する場合のコンテンション解消』を参照してください。

「拡張」タブの他の設定では、以下の項目を設定できます。

- 各ユーザー ID に 1 つの接続のみ許可
- 接続試行を中止するまでの秒数
- 切断試行を中止するまでの秒数

ユーザー定義文字 (UDC) を表示および印刷する場合の DBCS に関する考慮事項については、550 ページの『ユーザー定義文字の処理』を参照してください。

印刷

「**印刷**」タブでは、接続の印刷サポート設定を指定します。印刷サポートは、3270E、5250、および 5250W デフォルト接続でのみ使用可能です。

注: モバイル装置に対して最適化されているプロジェクトでは、印刷サポートは使用できず、「印刷」タブは表示されません。

印刷サポートが必要な場合は、「**印刷サポートを使用可能にする**」チェック・ボックスを選択します。

3270E 接続の印刷設定

3270E 接続の場合は、デフォルトの印刷設定は Adobe Portable Document Format (PDF) 設定です。PDF を使用する場合は、「**Adobe PDF ファイルのプロパティー**」セクションで、印刷ジョブに使用する用紙サイズ、ページの向き、およびフォントを選択できます。選択可能なフォントのリストは、使用する接続のコード・ページ設定によって異なります。コード・ページの詳細については、505 ページの『第 22 章 言語サポート』を参照してください。

その他のデフォルト以外の印刷設定を指定するには、「**初期化**」セクションと「**名前/値**」テーブルを使用できます。

「**初期化**」ボタンを使用して、ドロップダウン・ボックスから選択する印刷シナリオに基づいて印刷設定の起動設定を「**名前/値**」テーブルに追加します。起動設定として 3 つの印刷シナリオがあります。これらのシナリオと、シナリオにより生成される印刷設定を次に示します。

- デフォルト印刷 (Adobe PDF 形式)

名前	値
printDestination	false
printMimeType	application/pdf
printSaveAsExtension	.pdf
separateFiles	true
useAdobePDF	true
usePDT	false
useWindowsPrinter	false

- 基本テキスト・ファイル (プレーン・テキスト形式)

名前	値
PDTFile	/pdfpdt/basic.hodpdt
printDestination	false
printMimeType	text/plain
printSaveAsExtension	.txt
separateFiles	true
useAdobePDF	false
usePDT	true
useWindowsPrinter	false

- サーバーに接続されたデフォルトの Windows プリンターに送信 Web のみ

  または

  デフォルトの Windows プリンターに送信 RCP のみ

名前	値
printDestination	true
useAdobePDF	false
usePDT	false
useWindowsDefautlPrinter	true
useWindowsPrinter	true

「**初期化**」ボタンの使用はオプションです。ボタンをクリックした時点でテーブル内に値が存在している場合は、現行設定が置換されることを通知する警告メッセージが表示されます。クリックして操作を続行するか、または取り消すことができます。

各自の印刷設定を入力するか、または「**追加**」、「**編集**」、および「**除去**」ボタンを使用して、テーブルの既存の設定を変更できます。「**追加**」ボタンをクリックすると、 WebSphere Host On-Demand 印刷設定のリストのうち、サポートされている設定をすべてリストしたコンボ・ボックスが表示されます。

例えば、3270E PDF 印刷出力に対してさらに制御を加えるには、以下の設定を追加できます。

charsPerInch

1 インチ当たりの文字数。1 インチ当たりに印刷される文字の個数を指定します。Windows プラットフォームでは 3 つの選択肢 (10、12、および 17) が使用可能です。デフォルト値は 10 です。

linesPerInch

1 インチ当たりの行数。1 インチ当たりの行数を指定します。Windows プラットフォームでは 5 つの選択肢 (2、3、4、6、および 8) が使用可能です。デフォルト値は 6 です。

maxCharsPerLine

1 行当たりの最大文字数。1 行当たりの最大文字数を指定します。この値は、最大印刷位置や最大表示位置 (MPP) とも呼びます。1 から 255 の間の値を入力してください。デフォルト値は 80 です。

maxLinesPerPage

ページ当たりの最大行数。1 ページ当たりの最大行数 (上部マージンと下部マージンを含む) を指定します。この値は、最大ページ長 (MPL) とも呼ばれます。1 から 255 の間の値を入力してください。デフォルト値は 66 です。

RTLfile

双方向言語アプリケーション用の右から左方向のファイルの印刷。RTL 画面に表示されるとおりにファイルを印刷するには、「true」を指定します。デフォルトは「false」です。

注:

1. ユーザーのワークステーションに接続しているデフォルトの Windows プリンターへの印刷出力は、HATS リッチ・クライアント・アプリケーションでのみサポートされています。
2. WebSphere Application Server に接続しているデフォルトの Windows プリンターへの印刷出力は、HATS Web アプリケーションでのみサポートされています。
3. HATS 3270E の印刷では、印刷ジョブ内の未処理の印刷データを送信するための Transparency コマンドがサポートされません。
4. 日本語 JIS2004 サポートの詳細については、 512 ページの『3270E セッションの PDT 印刷および Print-to-File の JIS2004 サポート』を参照してください。

詳細については、 417 ページの『プロジェクト用印刷サポートの定義』を参照してください。

5250 および 5250W 接続の印刷設定

5250 接続および 5250W 接続で、**「印刷サポートを使用可能にする」**チェック・ボックスを選択した場合は、System i® Access for Web の「プリンター出力」ウィンドウの Web アドレスを指定する必要があります。デフォルトの Web アドレスは http://*hostname*/webaccess/iWASpool です。この *hostname* は、5250 または 5250W ホストの名前です。アプリケーションのユーザーは、「IWA プリンター出力 (IWA Printer Output)」ウィンドウで印刷オプションを設定できます。詳細については、417 ページの『プロジェクト用印刷サポートの定義』を参照してください。

画面処理

「**画面処理**」タブでは、接続起動時のブランク画面の処理方法と画面確定の処理方法を指定できます。これらの設定は、接続をデフォルトの接続として使用している場合だけに適用できます。

接続アプリケーション・イベントでデフォルト接続取得アクションが処理されると、HATS はホストへの接続を開くことを試行します。この接続試行が完全に成功しない場合は、アプリケーションが終了し、エラー・ページが表示されます。接続試行が完全に成功した場合は、このページに指定された画面タイマーに基づいて、ホスト画面を確定できるようになります。これらのタイマーは、カスタム画面認識イベント、ブランク画面アプリケーション・イベント (構成してある場合)、および一致しない画面イベントが完了したときのホスト画面の確定にも使用されます。画面設定のアルゴリズムと設定について詳しくは、567 ページの『付録 B. HATS 画面確定のリファレンス』を参照してください。

「**接続開始時にブランク画面を受信した場合**」設定では、ホストに正しく接続した後もホスト画面がブランクのままの場合に、どのように処理するかを選択できます。この場合、前述のとおり、ホスト時間に応じて画面を確定できます。以下のオプションを選択できます。

接続がタイムアウトになるまで待機してからエラーで終了する
このオプションを選択すると、ホストに正しく接続した後もホスト画面がブランクである場合に、HATS によってアプリケーションが終了され、エラー・ページが表示されます。

ブランク画面を表示する
このオプションを選択すると、ホストに正しく接続した後にホスト画面がブランクである場合でも、HATS は画面認識処理を実行します。

ユーザーに対してブランク画面が表示されないようにする場合は、ブランク画面を処理するためのイベント優先順位リストで画面認識イベントを指定するか、ブランク画面アプリケーション・イベントにアクションを追加してブランク画面を制御します。デフォルトでは、「一致しない画面」アプリケーション・イベントで、ユーザーに対しブランクのホスト画面が表示されます。画面認識イベントの詳細については、124 ページの『画面イベントの優先順位』を参照してください。アプリケーション・イベントの詳細については、125 ページの『アプリケーション・イベント』を参照してください。

ホスト・キーの送信
このオプションでは、ホストに正しく接続した後も接続がブランクのままである場合に、ホストにホスト・ファンクション・キーを送信します。このキーは一度だけ送信され、画面が確定され、HATS によって画面認識処理が進められます。ドロップダウン・リストから送信するキーを選択できます。このオプションは、ホストで常にブランクの初期画面が描画され、ホストで新規画面を表示する前にユーザーが特定のファンクション・キー (SysReq など) を実行依頼するしなければならない場合に便利です。

「**画面タイマー**」では、ホストが画面を HATS ランタイムに送信し終えるまでの待機時間の長さを構成できます。これらのタイマーの使用機会については、567 ページの『付録 B. HATS 画面確定のリファレンス』を参照してください。次の設定を指定できます。

初期ホスト画面を待機する最小時間

デフォルトは、2000 ミリ秒です。ホスト接続の準備ができてから、アプリケーションが初期画面の更新を待つ最小時間です。接続がしばらく作動可能状態になっているにもかかわらず、ホストからの初期画面の送信が遅い場合は、この値を大きくします。

画面が安定するまで待機する最大時間

デフォルトは、1200 ミリ秒です。初期画面の更新後に、アプリケーションが画面更新の到着を待つ最大時間です。ホストからのコンテンツ送信が遅く、画面を部分的に受信することが多い場合は、この値を大きくします。

画面が安定するまで待機する最大時間 (非同期更新を使用したセッションの場合)

この値は、アプリケーションが非同期更新 (リッチ・クライアント・アプリケーションの場合) または非同期更新アプレット (Web アプリケーションの場合) を使用している場合にのみ使用されます。詳しくは、134 ページの『非同期更新 RCP のみ』および 136 ページの『サーバー・プッシュ (アプレット) メソッドの使用』を参照してください。

初期デフォルト値は、400 ミリ秒です。ホストからの画面送信が遅く、画面を部分的に受信することが多い場合は、この値を大きくします。application.hap ファイルの com.ibm.common.AppletSettings クラスのブラウザー固有の設定が「**最新表示**」に設定されている場合、この値を 「**画面が安定するまで待機する最大時間**」の値より小さく設定することもできます。

セキュリティー

「**セキュリティー**」 タブには、Secure Sockets Layer (SSL) および Web 高速ログオン (WEL) の構成設定が表示されます。セキュリティー設定の詳細については、477 ページの『第 21 章 セキュリティーおよび Web 高速ログオン』を参照してください。

注: HATS EJB プロジェクトで接続に SSL を使用する場合は、「HATS EJB プロジェクト」ビューで EJB プロジェクトと「接続」フォルダーを開き、接続をダブルクリックします。その後、以下の指示に従います。HATS EJB プロジェクトの作成方法については、「*HATS Web アプリケーション・プログラマーズ・ガイド*」のセクション『HATS EJB アプリケーションの作成および使用』を参照してください。

SSL を使用可能にする

SSL を使用可能にするには、このチェック・ボックスを選択します。

注: Telnet サーバーで有効な既知の個人証明書が使用される場合は、必要な操作はこのボックスを選択する操作のみです。

PKCS12 鍵ストアをプロジェクトにインポート

PKCS12 鍵ストア・ファイルをプロジェクトにインポートするには、このオプションを選択します。「**インポート**」ボタンをクリックして鍵ストア・ファイルを見つけ、このファイルをプロジェクトにインポートします。この接続の構成で、インポートされた鍵ストア・ファイルを指すポインターが設定されます。接続構成から鍵ストア・ファイルを指すポインターを除去するには、「**除去**」ボタンをクリックします。鍵ストア・ファイル自体は、プロジェクトから

は除去されません。ファイルのインポートが完了すると、ファイル名が「**鍵ストア・ファイルへのパス**」編集ボックスに表示されます。これが必要な状況と PKCS12 鍵ストア・ファイルの作成方法の詳細については、477 ページの『SSL セキュリティーの使用可能化』を参照してください。

注: 鍵ストア・ファイルをインポートし、接続エディターで変更を保存した後に、鍵ストア・ファイルがプロジェクトに含まれるようにするため、プロジェクトを最新表示してください。プロジェクトを最新表示するには、「HATS Projects」ビューでプロジェクトを右クリックし、「**最新表示**」を選択します。プロジェクトの自動最新表示を設定するには、Rational SDP メニュー・バーで「**ウィンドウ**」>「**設定**」>「**その他**」>「**ワークスペース**」>「**自動で更新**」を選択します。

特定のパスで PKCS12 鍵ストアを使用

プロジェクトには組み込まないがターゲット・ランタイム・システムの任意の位置に保管する鍵ストア・ファイルを指定するには、このオプションを選択します。「**鍵ストア・ファイルへのパス**」編集ボックスに、ターゲット・システムでの鍵ストア・ファイルの完全なパスとファイル名を指定します。これが必要な状況と PKCS12 鍵ストア・ファイルの作成方法の詳細については、477 ページの『SSL セキュリティーの使用可能化』を参照してください。

注:

1. 開発システムでテスト中にこのファイルを使用するには、開発システム上で、ターゲット・ランタイム・システム内と同じ位置にこのファイルが保管されている必要があります。
2. HATS Web アプリケーションでは、プロジェクト .ear ファイル内に含まれない鍵ストア・ファイルを使用し、ターゲット WebSphere Application Server システムで Java 2 セキュリティーを使用する場合は、HATS アプリケーションが was.policy ファイルにアクセスする前に、WebSphere Application Server でこのファイルを更新しておく必要があります。was.policy ファイルは、META-INF ディレクトリーのプロジェクト .ear ファイルの「ナビゲーター」ビューに表示されます。例えば、鍵ストア・ファイルの読み取りアクセス権を付与するには、was.policy ファイルに以下のステートメントを追加します。

   ```
   permission java.io.FilePermission "c:¥¥myKeystores¥¥-", "read";
   ```

   myKeystores は、ターゲット WebSphere Application Server システムで鍵ストア・ファイルが保管されているフォルダーの名前です。詳しくは、497 ページの『Java 2 セキュリティー』を参照してください。

鍵ストア・ファイルへのパス

鍵ストア・ファイルをインポートした場合は、この編集ボックスにインポートしたファイルの名前が表示されます。「**特定のパスで PKCS12 鍵ストアを使用**」オプションを選択している場合は、この

編集ボックスに、ターゲット・ランタイム・システムでの鍵ストア・ファイルの完全なパスとファイル名を入力します。

パスワード

「**鍵ストア・ファイルへのパス**」編集ボックスに指定した鍵ストア・ファイルを開くために必要なパスワード。鍵ストア・ファイルを検出し、パスワードでこのファイルを開く操作をテストするには、「**検証**」ボタンを使用します。

注:

1. これは、鍵ストア・ファイルの作成時に使用されたパスワードと同じです。PKCS12 鍵ストア・ファイルの作成方法の詳細については、477 ページの『SSL セキュリティーの使用可能化』を参照してください。
2. プロジェクトに含まれていない鍵ストア・ファイルの位置とパスワードを検証するには、開発システム上で、ターゲット・ランタイム・システム内と同じ位置にこの鍵ストア・ファイルが保管されている必要があります。
3. パスワードは平文では保管されません。ただし HATS アプリケーションの展開後に、アプリケーションを再展開せずにパスワードを変更したい場合には、ランタイム・システムで接続を表す .hco ファイル内のパスワード・フィールドを変更できます。.hco ファイルを編集して変更した場合は、アプリケーションを再展開するまでは、パスワードが平文で保管されます。

Web 高速ログオンの使用 **Web のみ**

HATS Web アプリケーションでは、WEL を有効にして構成するには、このボックスを選択して「**構成**」ボタンをクリックします。詳しくは、481 ページの『Web 高速ログオン (WEL) の使用』を参照してください。

Kerberos サービス・チケットを使用してサインオンを自動化する (Windows ドメイン・クライアントのみ) **RCP のみ**

HATS リッチ・クライアント・アプリケーションでは、このボックスを選択して、Kerberos サービス・チケットを使用し、サインオンを自動化します。このサポートは、IBM i Kerberos ベース・ネットワーク認証とともに、基礎となる IBM Host On-Demand 接続ベースの自動化を使用します。詳しくは、Host On-Demand Knowledge Center において、接続ベースの自動化を構成する方法に関するセクション (http://publib.boulder.ibm.com/infocenter/hodhelp/v11r0/index.jsp?topic=/com.ibm.hod.doc/doc/logon/logon14.html) を参照してください。

注:

1. この設定は、5250 Telnet 接続のみで使用可能です。
2. この機能は、Windows ドメイン・クライアントで実行している場合にのみサポートされます。

プール

プールとは、初期状態で維持され、作成、初期化することなくいつでも使用できるホスト接続の 1 つのグループのことです。これにより、アプリケーションの開始、実行時の応答時間が短縮されます。

プールは、「**プール**」タブをクリックし、「**プールを使用可能にする**」チェック・ボックスを選択して使用可能にできます。プールが使用可能な場合には、接続が解放された後も、チェックイン画面定義で定義された許容状態にあると判断されれば、アクティブ接続のリストに維持されます。接続に対する新規要求が受信されると、プール内のアイドル接続の 1 つが使用されます。プール内にアイドル接続が存在しない場合は、最大接続数などの他のプール設定に基づいて、新しい接続が作成されることがあります。接続プールが使用不可の場合は、接続が解放された後、その接続は切断されます。

デフォルト接続上で画面間をナビゲートしたユーザーがチェックイン画面に戻る保証はないため、HATS デフォルト接続ではプールを使用しないようにお勧めします。統合オブジェクトで使用される、またはマクロ・トランザクションの実行アクションで使用されるバックグラウンド接続では、自動ナビゲーションでチェックイン画面に戻るようにプログラムできるので、プールを効率的かつ効果的に利用できます。

ユーザー自身の「**接続タイムアウト**」および「**接続限界**」を設定するオプションも用意されています。

- 「**接続タイムアウト**」の場合、「**接続を終了するまでの最大アイドル時間**」チェック・ボックス、「**接続を終了するまでの最大使用中時間**」チェック・ボックス、またはその両方のチェック・ボックスを選択してから、時間を秒単位で入力します。

  注:

  1. 「**接続を終了するまでの最大使用中時間**」は、プールの外部でも使用できます。
  2. 最大使用中時間には、1 つの HATS アプリケーション内のすべての接続に対して同じ値を設定します。
- 「**接続限界**」の場合、プール内に保持するアイドル接続の最小数またはプール内でアクティブにできる接続の最大数、あるいはこの両方をテキスト・ボックスに入力します。

  最大接続数に到達した後の処理を決定することもできます。該当するラジオ・ボタンを選択して新規 (非プール) 接続を作成するか、または「**プールされた接続を待機する最大秒数**」チェック・ボックスを選択し、テキスト・ボックスに待機時間を秒単位で入力して、使用可能なプール接続を待機および待機限界時間を設定することができます。接続待機時間を制限なしにする場合は、このチェック・ボックスをクリアします。

マクロ

「**マクロ**」タブには、接続マクロ、切断マクロ、およびチェックイン画面に関する情報が表示されます。両マクロとも、オプションです。この情報は、プールが使用できない場合でも、この 2 つのマクロが適用可能ため、プール情報とは別のタブ上にあります。

「**接続マクロ**」および「**切断マクロ**」の各ドロップダウン・リストから、接続マクロと切断マクロを選択します。

「**接続マクロ**」は、接続を最初に作成したときに、自動的に実行されます。接続マクロで許可されているプロンプトは、ユーザー・リストまたは Web 高速ログオンの使用に合致しているものだけです。プールが使用可能な場合、HATS サーバーの初期化時に接続マクロが実行されます。プールが使用不可の場合、HATS アプリケーションへの最初の要求が作成されたとき (通常は、接続イベント中) に接続マクロが実行されます。

注: ユーザーの信任状を使用して自動サインオン・マクロを実行する必要がある場合は、接続マクロを使用する代わりに、接続イベントにマクロ実行アクションを追加することができます。マクロの再生アクションを使用したマクロ実行では、プロンプトに対話式に応答したり、開始イベント中などに初期化されたグローバル変数を使用して対応したりできます。

「**切断マクロ**」は、接続が破棄されると、自動的に実行されます。切断マクロでは、プロンプトを使用できません。プールが使用可能な場合、HATS サーバーのシャットダウン時に切断マクロが実行されます。プールが使用不可の場合、HATS アプリケーションが接続を解放してシステムに戻すときに (通常は、切断イベント中) 切断マクロが実行されます。

「**チェックイン画面**」は、接続をプールに参加させるために経由する必要がある画面を表します。チェックイン画面は、プールが使用可能な場合だけに、使用できます。通常は、チェックイン画面は、接続マクロの終了画面と同じです。また、切断マクロの入力画面とも一致します。

チェックイン画面セクションは、「**プール**」タブでプールを使用可能にした場合に限り利用可能となります。プールが使用可能になると、チェックイン画面セクションには、以下のオプションが表示されます。

- 「**接続マクロの最初の終了画面を使用 (Use first exit screen of Connect macro)**」。これは、推奨されるオプションです。
- 「**接続マクロの特定の画面を使用 (Use a specific screen of Connect macro)**」。ここには、ドロップダウン・リストから画面マクロを選択するオプションが用意されています。
- 「**以前に取り込んだ画面に基づいて別の画面認識基準を使用**」により、ドロップダウン・リストからユーザーが選択した前に取り込まれた画面に基づいて、チェックイン画面に対する別の画面認識基準の使用が可能になります。

チェックイン画面に対して、認識基準を指定することもできます。画面認識基準の定義方法の説明については、175 ページの『画面認識基準または開始画面』を参照してください。

注: チェックイン画面を不明確に定義すると、ユーザーのプールが無駄になります。明確な認識基準を追加することお勧めします。

マクロの詳細については、381 ページの『第 11 章 マクロとホスト端末』を参照してください。接続マクロを使用すると、ユーザー・リストを設定して、接続の使用可能なユーザーのリストを指定することができます。

ユーザー・リスト

ユーザー・リストとは、それぞれがユーザー ID とパスワードを含むユーザー・プロファイルのリストのことです。ユーザー・リストは、ホスト接続と関連付けられた接続マクロにより使用できます。接続マクロが実行されると、ユーザー ID とパスワードがリストから引き出され、ホスト画面のユーザー ID とパスワードの両フィールドに設定されます。これにより、ユーザー ID とパスワードの事前定義リストが使用可能になり、ユーザーは、ユーザー ID およびパスワードを知る必要やホスト画面上で入力する必要がなくなります。

ユーザー・リストは、汎用ユーザー ID の集約リストを使用した自動化ログインが必要となる HATS Web アプリケーション、EJB、または Web サービスに多くのエンド・ユーザーが同時にアクセスする可能性がある場合に役立ちます。

HATS リッチ・クライアント・アプリケーションおよびそのユーザー・リストは、中央サーバーではなく、個々のエンド・ユーザー・システムに展開されるため、ユーザー・リストは通常あまり役に立ちません。これは、各展開において同時に同じユーザー ID を使用しようとし、その結果ログインに失敗する恐れがあるためです。リッチ・クライアント・アプリケーションでは、指定されたユーザー ID を同時に多くのユーザーが使用できるホスト・システムのユーザー・リストを正常に使用できる場合があります。これは、ユーザー・リストに、すべてのエンド・ユーザー・システムが同時に使用する単一の汎用ユーザー ID のみが提供される場合です。汎用ユーザー ID およびパスワードを提供し、パスワードを暗号化したい場合に、このような選択ができます。

接続エディターの「拡張」タブで、オプション「**各ユーザー ID ごとに 1 つの接続のみ許可**」が適切に設定され、ユーザー・リストに関連付けられたホスト・システムのタイプを反映するようにします。 このオプションが選択されていない場合、システムはホストへのすべての接続に対して 1 つのユーザー・プロファイルを再使用します。

「**ユーザー・リスト**」タブには、ユーザー・プロファイルのテーブルが表示され、ここで任意のユーザー・プロファイルを 1 つずつ追加、編集、または削除することができます。

「**追加**」をクリックすると、ユーザー ID、説明、およびパスワードを入力するようにプロンプトで要求されます。それらのフィールドはすべて必須であり、どのフィールドも、「**編集**」をクリックして変更できます。エントリーを削除することも、「**除去**」をクリックして行えます。

「**ユーザー・リスト・プロパティーの暗号化**」を選択し、暗号化された形式ですべてのユーザー・リスト項目のパスワードを保管します。パスワードは、「ソース」タブが選択されている場合、暗号化されて表示されます。暗号化されていない形式ですべてのパスワードを保管するには、このボックスをクリアします。パスワードは、常にアスタリスク (*) として表示され、「ユーザー ID」フィールドと「説明」フィールドは、この設定に関係なく、暗号化されないままです。

注: Host Publisher ユーザー・リストの暗号化については、 27 ページの『Host Publisher バージョン 4 からのマイグレーション』を参照してください。

「ユーザー ID」フィールドに達した際に、接続マクロを記録するように、ユーザー・リストからプロファイルの使用を指定するには、ホスト端末ツールバーの「**プロンプト・アクションを追加**」アイコンをクリックします。「プロンプト・アクションを追加」ページで、「**プロンプトをユーザー・リストからのプロパティーに設定**」を選択します。 「ユーザー・プロファイル」に、ドロップダウンからプロファイルを選択し、「ユーザー・リスト」プロパティーに、「**_userid**」を選択します。パスワード・フィールドに、同じプロセスを使用して、同じプロファイルを選択し、ユーザー・リスト・プロパティーに、「**_password**」を選択します。 マクロを記録中に選択したプロファイルは、マクロのテストに使用されるものです。リストのすべてのプロファイルは、ランタイムに必要に応じて使用されます。

接続マクロをホスト接続と関連付けるには、接続を編集します。接続マクロでは、「マクロ」タブで、ドロップダウンからご使用のマクロを選択します。

リストを使用する場合、切断マクロを記録して、ユーザー ID をログオフし、再使用できるようにする必要もあります。切断マクロをホスト接続と関連付けるには、接続を編集します。切断マクロでは、「マクロ」タブで、ドロップダウンからご使用の切断マクロを選択します。

クラスタリングとユーザー・リスト

HATS アプリケーションを実行して同じアプリケーション・ファイルを共有する WebSphere アプリケーション・サーバーのインスタンスを複数作成する場合、複数のログオン・ホストのユーザー・リストを使用できます。この場合、特別な考慮事項はありません。ただし、単一のログオン・ホストでユーザー・リストを使用する場合は、以下のような特別な考慮事項があります。

- アプリケーション関連ファイルが各アプリケーション・サーバーに物理的にコピーされておらず、メモリー内にあるだけの垂直クラスタリングでは、ユーザー・リストを使用できません。HATS を実行するすべてのアプリケーション・サーバーでユーザー・リストを排他的に使用する必要がありますが、実際には、ユーザー・リストが 1 つしかないために、この制限が生じます。
- アプリケーションの複数のコピーが別個のアプリケーション・サーバーで実行されている水平クラスタリングでは、それらのコピーが共通のユーザー ID を持たないようにするため、各アプリケーション・コピーでユーザー・リストを変更する必要があります。

ソース

「**ソース**」タブには、接続エディターで選択したか、またはデフォルトを受け入れた設定の多くについて、 .hco ファイルのタグと値が表示されます。接続エディターの他のタブを変更すると、「**ソース**」タブに表示されるタグと値もそれに合わせて変化します。

「**ソース**」タブのソースを編集することにより、.hco ファイルのタグと値を直接変更することもできます。この変更は、接続エディターの適切なタブに反映されます。

第 7 章 画面イベントの処理

HATS イベントは、到達する特定の状態に基づいて、一連のアクションを実行するリソースです。 HATS イベントには、アプリケーション・イベントと画面イベントの 2 種類があります。アプリケーション・イベントの詳細については、 125 ページの『アプリケーション・イベント』を参照してください。 画面イベントは、特定の画面認識基準と突き合わせることによってホスト画面が認識されると起動する HATS イベントです。画面イベントには、画面カスタマイズと画面組み合わせの 2 種類があります。

画面カスタマイズは、ホスト画面を認識したら一連のアクションを実行するように設計されている HATS 画面イベントです。画面カスタマイズの例としては、画面の認識、ユーザーの GUI への画面の変換、マクロの再生による画面のスキップなどがあります。画面カスタマイズ定義には、画面認識基準のセットと、ホスト画面が画面認識基準に一致した場合に実行するアクションのリストが含まれています。画面レベルのグローバル規則とテキスト置換設定も含まれます。新規の画面カスタマイズを作成するには、「画面カスタマイズの作成」ウィザードを使用してください。

画面組み合わせは、連続した類似のホスト画面から出力データを収集し、そのデータを結合して単一の出力ページ内に表示するように設計されている HATS 画面イベントです。画面組み合わせの例としては、部分的な出力データのみが含まれている画面を認識したり、残りのデータをすべて収集してユーザーに表示するために後続のすべての画面をナビゲートするなどがあります。画面組み合わせ定義には、組み合わせる開始画面と終了画面の画面認識基準のセット、画面間のナビゲート方法、および各画面から収集したデータを認識してレンダリングするために使用するコンポーネントとウィジェットが含まれています。また、画面カスタマイズと同様、実行するアクションのリスト、画面レベルのグローバル規則、およびテキスト置換設定も含まれています。 新規の画面組み合わせを作成するには、「画面組み合わせの作成」ウィザードを使用してください。

「画面カスタマイズの作成」ウィザード

HATS プロジェクトで、画面カスタマイズを定義するには、**「画面カスタマイズの作成」**ウィザードを使用します。画面カスタマイズ・ウィザードを使用する前に、ホスト端末を開いたままにしておくか、画面キャプチャーを用意しておく必要があります。BMS マップは、画面カスタマイズに使用する画面キャプチャーの作成にも使用できます。BMS マップ・セットの詳細については、 200 ページの『BMS マップ・セットのインポート』を参照してください。

次のいずれかのアクションを実行して、ウィザードにアクセスできます。

- **「HATS Projects」**ビューで、**「画面カスタマイズ」**フォルダーを右クリックし、**「新規 HATS」>「画面カスタマイズ」**を選択します。
- HATS ツールバーにある**「HATS 画面カスタマイズを作成」**アイコンをクリックします。

© Copyright IBM Corp. 2003, 2015

- ホスト端末ツールバーにある「**HATS 画面カスタマイズを作成**」アイコンをクリックします。
- 「**HATS Projects**」ビューで、プロジェクトの「**画面キャプチャー**」フォルダーを開きます。キャプチャー画面をダブルクリックし、エディターで開きます。 エディターが開いたら、「**HATS 画面カスタマイズを作成**」アイコンをクリックします。

ウィザードで使用可能な設定の詳細については、175 ページの『画面イベントの編集』を参照してください。

画面カスタマイズを対象に定義した設定は、画面カスタマイズ (.evnt) ファイルに保存されます。このファイルを参照するには、「**HATS Projects**」ビューの「**画面カスタマイズ**」フォルダーを展開します。 画面カスタマイズ・ファイルの表示および変更を行う場合は、画面イベント・エディターを使用してください。

「画面組み合わせを作成」ウィザード

HATS プロジェクトで、画面組み合わせを定義するには、「**画面組み合わせを作成**」ウィザードを使用します。「画面組み合わせを作成」ウィザードを使用する前に、ホスト端末か画面キャプチャーを開いておく必要があります。 BMS マップは、画面組み合わせに使用する画面キャプチャーの作成にも使用できます。 BMS マップ・セットの詳細については、200 ページの『BMS マップ・セットのインポート』を参照してください。

次のいずれかのアクションを実行して、ウィザードにアクセスできます。

- 「**HATS Projects**」ビューで、「**画面組み合わせ**」フォルダーを右クリックし、「**新規 HATS**」>「**画面組み合わせ**」をクリックします。
- HATS ツールバーにある「**HATS 画面組み合わせを作成**」アイコンをクリックします。
- ホスト端末ツールバーにある「**HATS 画面組み合わせを作成**」アイコンをクリックします。
- 「**HATS Projects**」ビューで、プロジェクトの「**画面キャプチャー**」フォルダーを展開します。キャプチャー画面をダブルクリックし、エディターで開きます。エディターが開いたら、「**HATS 画面組み合わせを作成**」アイコンをクリックします。

ウィザードで使用可能な設定の詳細については、175 ページの『画面イベントの編集』を参照してください。

画面組み合わせを対象に定義した設定は、画面組み合わせ (.evnt) ファイルに保存されます。このファイルを参照するには、「**HATS Projects**」ビューの「**画面組み合わせ**」フォルダーを展開します。 画面組み合わせファイルの表示および変更を行う場合は、画面イベント・エディターを使用してください。

注: HATS Dojo ウィジェットは、画面組み合わせではサポートされていません。

画面イベントの編集

画面イベント・エディターは、画面カスタマイズ .evnt ファイルまたは画面組み合わせ .evnt ファイルの名前をダブルクリックすれば起動できます。以下のセクションでは、画面イベント・エディターの各タブについて説明します。

概要

画面イベント・エディターの「**概要**」タブには、画面イベントを作成したときに指定したすべての情報が要約されています。このタブには、画面イベントの名前と説明、画面認識基準の作成に使用された画面の名前と画像、実行されるアクションの要約、および画面認識基準の要約が収容されています。画面組み合わせの場合、このタブには、ナビゲーション、コンポーネント、終了画面認識基準設定の要約も収められています。 このタブで、画面イベントの説明を変更したり、画面イベントに関連付ける別の画面を選択したりすることができます。選択した画面は、画面認識基準の変更やアクションの追加など、画面イベントに変更を加えるときに使用されます。

画面認識基準または開始画面

画面カスタマイズ・イベントの編集時には、「**画面認識基準**」タブが表示されます。 画面組み合わせイベントの編集時には、「**開始画面**」タブが表示されます。いずれの場合にも、タブには、画面イベントを起動するホスト画面に一致するように設定した画面認識基準が表示されます。基準を組み合わせてホスト画面を認識させることができます。基準には、画面上のフィールドまたは入力フィールドの総数、カーソル位置の座標、および画面上の定義済み長方形の内部または画面上の任意の場所にあるテキスト・ストリングなどが含まれます。このタブで、画面認識基準を追加、編集、または削除できます。

注:

DBCS サポートを使用する場合の考慮事項については、 547 ページの『画面認識基準/開始画面』を参照してください。

関連する画面キャプチャー

このオプションが表示されるのは、画面組み合わせイベントの「**開始画面**」タブの場合に限られます。 別の画面キャプチャーを選択して画面組み合わせの開始画面と関連付けるには、「**変更**」ボタンをクリックします。

フィールド基準

画面の「**フィールドの総数**」、画面の「**入力フィールドの数**」、またはこれら両方を画面認識基準として使用できます。「**フィールドの総数**」設定には、入力フィールド、保護テキスト・フィールド、および隠しフィールドが含まれます。これらは、このタブに表示される最初の 2 つの基準です。

これらの基準のチェック・ボックスを選択すると、画面の認識に使用されます。各フィールド基準の隣のフィールドは、「**概要**」タブで指定したフィールドと入力フィールドの総数を示します。この画面イベントに使用される画面を変更する場合、「**最新表示**」をクリックして、選択した画面の値を更新してください。

注: ある数のフィールドを持つ画面を認識するためにフィールド基準を使用し、別の画面にそれと同じ数のフィールドが含まれていない場合は、別の画面は認識されません。例えば、ある画面に 10 フィールドを持つ 10 ファイルのリストがあるとします。ホストが、リスト内に 8 ファイル、および 8 フィールドのみを持つ画面を表示する場合は、2 番目の画面は、最初の画面に一致した画面イベントのフィールド数の基準に一致しません。

「**オプション**」および「**反転**」チェック・ボックスについては、『オプションおよび非オプションの画面認識基準』および 177 ページの『画面認識基準の反転一致』を参照してください。

カーソル位置基準

カーソルの初期位置を画面認識基準として、単独、または他の基準とともに使用するには、「**カーソル位置**」チェック・ボックスを選択します。カーソル位置基準の隣のフィールドは、「**概要**」タブで指定した画面のカーソル位置の行と列を示します。この画面イベントに使用される画面を変更する場合、「**最新表示**」をクリックして、選択した画面の初期カーソル位置の行と列の値を更新してください。

「**オプション**」および「**反転**」チェック・ボックスについては、『オプションおよび非オプションの画面認識基準』および 177 ページの『画面認識基準の反転一致』を参照してください。

オプションおよび非オプションの画面認識基準

設定した画面認識基準がオプションか非オプションかを選択できます。「**オプション**」チェック・ボックスを選択しない場合は、認識基準は非オプションと見なされます。「**オプション**」チェック・ボックスの使用方法は、Host On-Demand 画面記述子属性 **optional** に対応して決まります。

オプションの認識基準と非オプションの認識基準の両方がある場合、HATS は最初に非オプションの基準をチェックします。すべての非オプションの認識基準が一致した場合、画面が一致したことになります。少なくとも 1 つの非オプション認識基準が一致しない場合、HATS はオプションの認識基準をチェックします。画面が基準に一致するためには、すべての非オプション基準または少なくとも 1 つのオプション基準が HATS で見つかる必要があります。それ以外の場合は画面は一致しません。次の例で、この考え方について詳しく説明します。

注: 非オプションは必須という意味ではありません。

次の例に示す値を使用して、カーソル位置と 2 つのテキスト・ストリングを定義したとします。

```
Cursor position recognition  Optional
                             Row: 1   Column: 1

String recognition           Non-optional
                             String 1: Welcome
                             Start position: Row: 1   Column: 6
                             End position:   Row: 1   Column: 12

                             String 2: Username
                             Start position: Row 20   Column 10
                             End position:   Row 20   Column 17
```

この例では、画面が一致するためには、HATS で両方のテキスト・ストリング、またはカーソル位置が見つかる必要があります。HATS は非オプションの基準を最初にチェックするので、テキスト・ストリングがまず検索されます。HATS がホスト画面の指定領域内で両方のテキスト・ストリングを見つけることができない場合、オプションの基準 (カーソル位置) を見つけることができるかどうかをチェックします。

画面認識基準の反転一致

設定した画面認識基準がホスト画面に一致するか一致しないかを選択できます。「**反転**」チェック・ボックスを選択すると、認識基準が画面に一致しない場合に基準が「true」となります。

逆に、「**反転**」チェック・ボックスをクリアすると、認識基準が画面に一致した場合に基準が「true」となります。

「**反転**」チェック・ボックスの使用方法は、画面記述子属性 **invertmatch** に対応して決まります。

追加の基準

ストリング基準: テキスト・ストリング位置基準は、タブの「追加の基準」セクションに表示されます。 画面認識基準としてストリング位置を設定している場合は、その位置がテーブルに示されます。このテーブルでは、テキストが含まれる画面位置のタイプ、選択したテキストの文字の一部、テキストの大文字と小文字を区別するか、テキストはオプションか、またはテキストを反転するかが示されます。

ストリング基準を定義するテーブルの行を強調表示して「**編集**」をクリックするか、「**追加**」をクリックすると、「画面基準 (Screen Criterion)」ダイアログが表示されます。 このダイアログ・パネルで、テキスト・ストリング情報を変更または編集できます。ストリングの属性 (大文字小文字の区別、オプション、または反転) を選択することもできます。このパネルには、「**概要**」タブで選択した画面が表示されます。

テキストの周囲に長方形を描いて、画面上の任意のテキストを選択できます。カーソルを画面上の一点に置き、左マウス・ボタンをクリックして押しながら別の点に移動して長方形を描きます。ダイアログの右側のフィールドに、選択したテキスト、および長方形の開始点および終了点の行とカラムの数値が表示されます。「**画面の任意の場所**」、「**指定位置**」、または「**長方形の領域内**」のいずれかのラジオ・ボタンをクリックして、テキストを含む画面部分を指定できます。選択したテキストを画面認識基準にマッチングさせるときに、大文字と小文字の区別が必要な場合は、「**大文字と小文字を区別**」チェック・ボックスを選択します。

「**オプション**」および「**反転**」チェック・ボックスについては、176 ページの『オプションおよび非オプションの画面認識基準』および 『画面認識基準の反転一致』を参照してください。

選択を終了したら、「**OK**」をクリックします。

グローバル変数基準: グローバル変数を使用して画面認識基準を定義するには、「**追加**」ボタンの横にある**下矢印**をクリックして、「**グローバル変数基準**」を選択します。 グローバル変数基準を使用する場合は、「**グローバル変数**」ドロップダウ

ン・リストからグローバル変数を指定するか、値を入力する必要があります。以下の 5 項目の 1 つを選択して、「**検証ロジック**」を指定します。

- **このグローバル変数は存在する**
- **このグローバル変数は存在しない**
- **このグローバル変数の長さを検査** (この場合、長さの比較方法をドロップダウン・リストから指定し、比較する長さを指定できます)
- **このグローバル変数の整数値を検査** (この場合、整数値の比較方法をドロップダウン・リストから指定し、比較する整数値を指定できます)
- **このグローバル変数のストリング値を検査** (この場合、ストリング値の比較方法をドロップダウン・リストから指定し、比較するストリング値を指定できます)

「**属性**」セクションの「**オプション**」および「**反転**」チェック・ボックスについては、176 ページの『オプションおよび非オプションの画面認識基準』 および 177 ページの『画面認識基準の反転一致』を参照してください。

拡張グローバル変数のプログラミングの詳細については、ご使用のアプリケーション環境に応じて、「*HATS Web アプリケーション・プログラマーズ・ガイド*」または「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

色基準: 「**追加**」ボタンの横にある下矢印をクリックして「**色基準**」を選択することにより、色による画面認識基準を定義できます。これにより、パネルがオープンし、行およびカラムの位置と、前景色および背景色を選択できます。

「**オプション**」および「**反転**」チェック・ボックスについては、176 ページの『オプションおよび非オプションの画面認識基準』および 177 ページの『画面認識基準の反転一致』を参照してください。

領域と値の比較の基準: 「**領域と値の比較**」**基準**により、ホスト画面の選択領域にあるテキストとハードコーディングされた値を比較できます。

まず、端末の領域を枠で囲むか、「**領域の定義**」セクションで認識対象とする座標を入力します。テキスト・ボックスに「**端末上の現行値**」も表示されます。

次に、領域の比較対象および比較方法を指定できます。「**比較の定義**」セクションで、「**領域**」の隣にあるドロップダウン・リストから領域の比較方法を選択し、「**値**」テキスト・ボックスに比較対象を入力します。

「**比較タイプ**」は、「**数値**」または「**テキスト**」のいずれかです (「テキスト」の場合は「**大文字と小文字を区別**」 チェック・ボックス・オプションもあります)。

属性も定義できます。「**オプション**」および「**反転**」チェック・ボックスについては、176 ページの『オプションおよび非オプションの画面認識基準』および 177 ページの『画面認識基準の反転一致』を参照してください。

選択を終了したら、「**OK**」をクリックします。

領域を領域と比較する基準: 「**領域を領域と比較する**」**基準**により、ホスト画面の特定の領域を同じホスト画面の別の領域と比較できます。

まず、「**最初の領域**」タブを選択し、端末の領域を枠で囲むか、「**領域の定義**」セクションで認識対象とする座標を入力します。テキスト・ボックスに「**端末上の現行値**」も表示されます。

次に、「**比較の定義**」セクションで、「**領域 1**」と「**領域 2**」の間にあるドロップダウン・メニューから、これらの領域の比較方法を指定します。

「**比較タイプ**」は、「**数値**」または「**テキスト**」のいずれかです (「テキスト」の場合は「**大文字と小文字を区別**」 チェック・ボックス・オプションもあります)。

属性も定義できます。「**オプション**」および「**反転**」チェック・ボックスについては、176 ページの『オプションおよび非オプションの画面認識基準』および 177 ページの『画面認識基準の反転一致』を参照してください。

「**2 番目の領域**」タブについても同じステップを繰り返し、完了したら「**OK**」をクリックします。

条件基準: 「**条件基準**」を使用すると、マクロ・ランタイムが画面認識時に評価する条件式 (例えば、`$intNumVisits$ == 0`) を指定できます。

画面認識時にマクロ・ランタイムは、条件式を評価し、ブール結果を得ます。

条件式が true に評価される場合、マクロ・ランタイムはこの記述子を true と評価します。等しくない場合、マクロ・ランタイムはこの記述子を false として評価します。

「**条件基準**」 は、マクロ・ランタイムが、1 つ以上の変数の値、または Java メソッドの呼び出しの結果に基づいて、次に処理されるマクロ画面を判別できるようにすることによって、画面認識の柔軟性と能力を向上させます。

「**条件**」入力フィールドに、マクロ・ランタイムが画面認識時に評価する条件式を入力します。

「**オプション**」および「**反転**」チェック・ボックスについては、176 ページの『オプションおよび非オプションの画面認識基準』および 177 ページの『画面認識基準の反転一致』を参照してください。

カスタム基準: 「**カスタム基準**」を使用して、カスタム記述コードを呼び出すことができます。

「**クラス名**」入力フィールドに、実行するメソッドを含む Java クラスの名前を指定するか、または「**参照**」をクリックして **Source** ディレクトリーからクラス名を選択します。

「**オプション**」および「**反転**」チェック・ボックスについては、176 ページの『オプションおよび非オプションの画面認識基準』および 177 ページの『画面認識基準の反転一致』を参照してください。

レンダリング (画面組み合わせのみ)

「**レンダリング**」タブは、画面組み合わせイベントの場合にのみ表示されます。このタブは、画面組み合わせによって収集する情報を認識するためのコンポーネントを設定する場合に使用します。 また、このタブは、コンポーネントをレンダリング

するときに使用するウィジェットを設定する場合にも使用します。画面組み合わせイベントでは、HATS コンポーネントとウィジェットのサブセットがサポートされています。 HATS コンポーネントおよびウィジェットの使用の詳細については、213 ページの『ホスト・コンポーネントを挿入』を参照してください。

「**領域の変更**」ボタンをクリックすると、画面キャプチャーの領域と、コンポーネントによってみなされる画面の領域を変更できます。

ポートレット以外のプロジェクトでは、以下の設定を使用すると、画面組み合わせの結果がユーザーに対して初めて表示される時刻を、画面組み合わせが完了する時刻と関連させて制御することもできます。

動的キャッシュ・コンテンツ・ロードを使用する Web のみ

HATS が画面組み合わせのすべての画面をナビゲートし、すべての結果を組み合わせてから結果をユーザーに表示するように指定する場合は、このボックスを未選択のままにします。最初の画面からの結果が収集されたらすぐに、ユーザーに対する結果の表示が HATS によって開始され、画面がナビゲートされるにつれて結果の表示を更新するように指定する場合は、このボックスを選択します。

ナビゲーション (画面組み合わせのみ)

「**ナビゲーション**」タブは、画面組み合わせイベントの場合にのみ表示されます。このタブには、画面組み合わせの開始画面から中間画面を経由して終了画面までナビゲートするために必要なコマンドが表示されます。「**追加**」、「**編集**」、「**除去**」の各ボタンを使用して、「**上へ移動**」および「**下へ移動**」の 2 つのナビゲーション方向のコマンドを設定します。表示される「**追加コマンド**」ダイアログまたは「**編集コマンド**」ダイアログには、3 つのコマンド・オプションがあります。ナビゲーション方向ごとに複数のコマンドを設定するには、「**追加**」ボタンを複数回クリックします。設定できる 3 つのコマンド・オプションを以下に示します。

キーを送信

このオプションを使用して、ホスト・システムに送信するホスト・キーを選択できます。

テキストを挿入

このオプションは、現在のカーソル位置にテキストを挿入するときに使用します。

カーソル位置を設定

このオプションは、現行カーソル位置をホスト画面に設定するときに使用します。

終了画面 (画面組み合わせのみ)

「**終了画面**」タブは、画面組み合わせイベントの場合にのみ表示されます。このタブには、画面組み合わせの終了を定義している、設定した基準が表示されます。このタブで、画面認識基準を追加、編集、または削除できます。「**反復カウントのみ**」、「**画面記述子のみ**」、「**反復カウントまたは画面記述子のいずれかの最初の一致**」という 3 種類の終了基準のいずれかを選択できます。

終了基準を選択

反復カウントのみ
この基準を「**反復の回数**」設定と組み合わせて選択すると、この画面組み合わせが固定の画面数の後に終了することを指定できます。

画面記述子のみ
この基準を他の画面認識設定 (「**フィールドの総数**」、「**入力フィールドの数**」、「**カーソル位置**」、「**追加の基準**」など) と組み合わせて選択すると、画面組み合わせの終了画面を定義できます。

反復カウントまたは画面記述子のいずれかの最初の一致
この基準をこれ以外のすべての設定と組み合わせて選択すると、この画面組み合わせが終了するタイミングを、固定の画面数の表示後と、終了画面が認識されたときのうちいずれか早い方に指定できます。

反復回数

「ナビゲーション」タブに定義されているナビゲーションの実行回数を設定する場合は、このボックスを選択します。この回数に到達すると、終了画面に到達したものとみなされます。

関連する画面キャプチャー

別の画面キャプチャーを選択して画面組み合わせの終了画面と関連付けるには、「**変更**」ボタンをクリックします。

画面記述子の設定

終了画面の画面記述子設定は、開始画面の認識基準設定と同じです。これらの設定の説明については、175 ページの『画面認識基準または開始画面』を参照してください。

アクション

画面イベント・エディターの「**アクション**」タブには画面イベントの構成済みアクションが表示されます。

アクションの適用順序を変更するには、順序を変更するアクションをリスト内で選択し、**上矢印**または **下矢印**をクリックして上または下に移動します。

リストに表示されるすべてのアクションを「**編集**」または「**除去**」できるだけでなく、新しいアクションを「**追加**」することもできます。

これらのアクションは、リストされている順に、HATS ランタイムによって実行されます。例えば、一部のフィールドに値が事前に入力された変換済みホスト画面をユーザーに対して表示するには、ホスト画面を変更してから (「**データを挿入**」アクションを使用するなど)、変換済みホスト画面を表示します (「**変換を適用**」 アクションを使用)。画面イベントで「**マクロを実行**」アクションまたは「**キーを送信**」アクションを使用する場合、そのアクションはリスト内の最後のアクションとする必要があります。

画面イベント内のすべてのアクションの実行が完了すると、HATS は変換ホストにデフォルトのコマンドを送信して、現在認識されている画面からホスト・アプリケーションをクリアします。この処理を行わないと、画面イベント内のアクションで

無限ループが発生する可能性があります。「**アクション・リストの完了時にホスト・キーを送信して画面を変更**」を選択すると、アクション・リストの完成後に送信するキーを判別できます。デフォルトでは、HATS ランタイムによって、アクション・リストの終わりに [enter] が変換ホストに送信されます。エイド・キーは、以下の条件をすべて満たす場合に送信されます。

- 「**変換を適用**」アクションを実行しても、サーブレットがコマンド・パラメーターを受信しない。
- 「**マクロを実行**」アクションを使用して、変換ホストに対してマクロが実行されていない。
- 「**キーを送信**」アクションによって、変換ホストに画面変更エイド・キーが送信されていない。

アクション・リストに「変換を適用」アクションが含まれていて、変換への応答として F2 コマンドを送信した場合、デフォルトのコマンドではなく、[pf2] コマンドが送信されます。同様に、「**マクロを実行**」アクションを使用して画面イベントを終了した場合もデフォルトのコマンドは送信されません。

このタブで、アクションを追加、編集、または削除できます。指定可能なアクションを以下に示します。

- 変換を適用
- ビジネス・ロジックを実行
- グローバル変数を抽出
- データを挿入
- グローバル変数を設定
- グローバル変数を除去
- グローバル変数を送信 (HATS 標準ポートレット・プロジェクトのみ)
- URL または SWT コンポジットを表示 RCP のみ
- URL を表示 Web のみ
- URL に転送 Web のみ
- マクロを実行
- マクロ・トランザクションを実行
- キーを送信
- 切断
- 一時停止

画面イベントに定義したすべてのアクション・タイプとその説明は、「**アクション**」タブのテーブルに表示されます。テーブルの行を強調表示して「**編集**」をクリックすると、「**アクションを編集**」ダイアログが表示されます。

「**追加**」をクリックすると、「アクションを選択」ウィザードが表示されます。最初のパネルには、すべての使用可能なアクションのリストが表示されます。実行するアクションを選択するには、そのアクションのラジオ・ボタンをクリックし、「**次へ**」をクリックします。選択したアクションに応じて、パネルのほかの部分に、そのアクションに対して指定できる情報が表示されます。

注: 「**アクションを編集**」ダイアログでアクション・タイプを変更することはできません。変更できるのはアクションに適用する情報のみです。

「変換を適用」アクション

この画面イベントのアクションとして変換を適用することにした場合、HATS Web プロジェクトでは、プロジェクトで定義した変換のドロップダウン・リストから、適用する変換を選択できます。

HATS リッチ・クライアント・プロジェクトの場合は、「**参照**」ボタンを使用して、現行プロジェクトのクラス・パスにある画面変換のリストを表示できます。このリストの対象になるのは、通常、現行プロジェクトの変換のみですが、現行プロジェクトによって参照される他のプラグインを基にした他の変換や、現行プロジェクトによって参照される .jar ファイル内に格納されている変換を対象にすることもできます。

詳しくは、 203 ページの『第 8 章 変換の操作』を参照してください。

「**テンプレート**」フィールドには、デフォルトで選択される「**(デフォルト・テンプレート)**」があります。この変換に適用する別のテンプレートを選択しない限り、GUI で変換を囲むテンプレートは、プロジェクトのデフォルト・テンプレートとして指定したテンプレートとなります。ドロップダウン・リストに、プロジェクトで指定したすべてのテンプレートが含まれます。

ユーザーが押したホスト・キーが、すべてのアクションの実行が完了するまで待機するのではなくすぐにホストに送信されるようにするには、「**即時ホスト・キー**」をクリックします。ホストに即時送信するキーのチェック・ボックスを選択します。ホスト・キーが即時に送信されると、その他のアクションは実行されなくなります。 これらのキーの即時送信は、現在の変換にのみ適用され、プロジェクト内のすべての変換には適用されません。

プロジェクト・レベルおよび画面レベルのグローバル規則処理を使用不可に設定するには、「**プロジェクト・レベルおよび画面レベルのグローバル規則を適用**」チェック・ボックスのチェック・マークを外します。

「ビジネス・ロジックを実行」アクション

あるビジネス・ロジックをこの画面イベントのアクションとして実行する場合は、実行するビジネス・ロジックの完全修飾 Java クラス名と Java メソッドを所定のフィールドに指定する必要があります。「**クラス名**」フィールドの隣の「**ブラウズ**」をクリックして、ビジネス・ロジック・メソッドが定義されているクラスを選択できます。HATS Toolkit の「**HATS Projects表示**」タブにある「**ソース**」フォルダーの下に定義されている任意のクラスを選択できます。このビジネス・ロジック用の Java コードを作成していない場合は、「**HATS Projects**」表示タブを右クリックし、「**新規 HATS**」>「**ビジネス・ロジック**」をクリックして、「ビジネス・ロジックの作成」ウィザードを起動します。

ビジネス・ロジックの詳細については、ご使用のアプリケーション環境に応じて、「*HATS Web アプリケーション・プログラマーズ・ガイド*」または「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

「グローバル変数を抽出」アクション

ホスト画面から情報を抽出して、グローバル変数として定義することができます。

ホスト画面領域には、グローバル変数として割り当てる画面領域の開始および終了の行とカラムを定義します。

「**拡張**」セクションで、「**抽出する平面**」ドロップダウン・リストを使用して、抽出するデータ平面を選択してください。以下のオプションがあります。

- テキスト
- 色
- フィールド
- 拡張フィールド
- DBCS
- グリッド

注: ホスト・アクセス・クラス・ライブラリー (HACL) 表示スペース・モデル内のさまざまなデータ平面の形式および内容については、『ホスト・アクセス・クラス・ライブラリー・プレーン -- 形式および内容』(http://www-01.ibm.com/support/knowledgecenter/SSS9FA_11.0.0/com.ibm.hod.doc/doc/hacl/DWYL0M88.HTML) を参照してください。

グローバル変数のデータを抽出する際には、新しいグローバル変数の名前を指定するか、既存のグローバル変数名を「**名前**」フィールドのドロップダウン・リストから選択することができます。

ホスト画面の複数行から抽出されたテキストをグローバル変数内に定義する方法を指定するには、「**拡張**」をクリックします。いずれかのラジオ・ボタンを選択して、1 つのストリングとして抽出するか、複数ストリングのリスト (索引付き) として抽出するかを指定する必要があります。「**この領域を 1 つのストリングとして抽出**」オプションを選択すると、抽出されたデータは単一オブジェクトとしてグローバル変数に、2 次元文字配列の形式で保存されます。「**この領域をストリングのリストとして抽出**」オプションを選択すると、データは 2 次元文字配列として抽出され、個別の 1 次元配列に分割されます。配列はそれぞれ、抽出された単一行を表し、グローバル変数内の索引に格納されます。

「**拡張**」をクリックする前に「**名前**」フィールドで既存のグローバル変数を選択した場合は、以下のいずれかのラジオ・ボタンを選択して HATS での抽出データの処理方法を指定する必要があります。

- **この新しい値で既存の値を上書き**
- **指定の索引から開始して既存の値をこの新しい値で置換**
- **この新しい値を既存の値の最後の索引の後に追加**
- **指定の索引でこの新しい値を既存の値に挿入**

指定の索引を使用する 2 つのオプションでは、「**索引**」フィールドに索引番号を入力する必要があります。

次の例に、選択するオプションにより変数値がどのように変更されるかを示します。まず、「sample」という既存の索引付き変数について説明します。「sample」の

値は「a b c d」です。この値のうち、「a」の索引は 0 であるため、「sample[0]」の値は「a」になり、「b」の索引は 1 であるため、「sample[1]」の値は「b」になります (以下同様)。新しい値のセット「e f g」を抽出するとします。

- 「**この新しい値で既存の値を上書き**」ラジオ・ボタンをクリックすると、「sample」の「a b c d」の値が「e f g」に変わります。
- 「**指定の索引から開始して既存の値をこの新しい値で置換**」ラジオ・ボタンをクリックして索引 2 を使用すると、「sample」の「a b c d」の値が「a b e f g」に変わります。
- 「**この新しい値を既存の値の最後の索引の後に追加**」ラジオ・ボタンをクリックすると、「sample」の「a b c d」の値が「a b c d e f g」に変わります。
- 「**指定の索引でこの新しい値を既存の値に挿入**」ラジオ・ボタンをクリックし、索引 2 を想定すると、「sample」の「a b c d」の値が「a b e f g c d」に変わります。

「**共用**」チェック・ボックスを選択すると、グローバル変数が共用されるようになります。

グローバル変数の詳細については、 399 ページの『第 12 章 グローバル変数との対話』を参照してください。

「データを挿入」アクション

データの挿入先にするホスト画面の領域を設定する場合は、マウスを使用するか、「**開始行**」および「**開始桁**」の設定を使用します。 ホスト画面の特定のフィールドを強調表示するには、「**フィールドを強調表示**」の隣にある別のオプションを選択します。入力フィールドが画面上で定義されている位置を表示するには、「**入力**」チェック・ボックスを選択します。保護フィールドを表示する場合は、「**保護**」チェック・ボックスを選択します。隠しフィールドを強調表示したい場合は、「**非表示**」チェック・ボックスを選択します。入力フィールド、保護フィールド、または隠しフィールドの強調表示色を変更する方法については、 149 ページの『HATS プリファレンスの使用』を参照してください。

「**次へ**」をクリックし、適切なラジオ・ボタンをクリックしてデータがストリングであるかグローバル変数であるかを選択します。ストリングを挿入するには、提供された入力フィールドにテキストを入力します。グローバル変数を挿入するには、既存のグローバル変数の名前をドロップダウン・リストから選択するか、グローバル変数の名前を入力します。異なるアプリケーションからの共用グローバル変数を使用する場合は、名前をフィールドに入力する必要があります。

グローバル変数の値に索引が付いている (ストリングのリストが含まれる) 場合は、「**拡張**」をクリックします。ラジオ・ボタンのいずれかをクリックして、指定した場所にすべてのストリングを並べて挿入するか、または画面の長方形領域に個別の行としてストリングを挿入するかを選択する必要があります。

「**共用**」チェック・ボックスを選択すると、グローバル変数が共用されるようになります。

グローバル変数の詳細については、 399 ページの『第 12 章 グローバル変数との対話』を参照してください。

注: ホスト画面への情報の挿入は、グローバル変数の変換が実行されて GUI に表示される*前に* 行う必要があります。アクションの順序の変更方法については、181 ページの『アクション』を参照してください。

「グローバル変数を設定」アクション

プロジェクト内の他のオブジェクトのほか、.ear の他のプロジェクトで使用するグローバル変数を設定できます。グローバル変数を設定する際、「**名前**」フィールドに名前を指定するか、ドロップダウン・リストから既存のグローバル変数名を選択できます。既存の索引付きグローバル変数を選択した場合、「**拡張**」をクリックして、設定値の処理方法を指定する必要があります。以下のオプションを使用できます。

- **この新しい値で既存の値を上書き**
- **指定の索引から開始して既存の値をこの新しい値で置換**
- **この新しい値を既存の値の最後の索引の後に追加**
- **指定の索引でこの新しい値を既存の値に挿入**

指定の索引を使用する 2 つのオプションでは、「**索引**」フィールドに索引番号を入力する必要があります。

選択したオプションに基づいて変数値を設定する方法の例については、184 ページの『「グローバル変数を抽出」アクション』を参照してください。

グローバル変数を固定値に設定する場合は、入力フィールドに値を入力します。

グローバル変数を計算値に設定する場合は、使用するオペランドおよび計算オペレーションを指定します。オペランドは、フィールドに固定値として入力するか、または既存のグローバル変数値を計算に使用できます。既存の索引付きグローバル値を使用し、使用する変数の索引をオペランドとして指定する場合は、「**拡張**」をクリックします。「**索引**」フィールドに索引の数を入力します。

「**共用**」チェック・ボックスを選択すると、グローバル変数が共用されるようになります。

グローバル変数の詳細については、399 ページの『第 12 章 グローバル変数との対話』を参照してください。

「グローバル変数を除去」アクション

このアクションを使用して、1 つ以上のグローバル変数を除去します。1 つのローカルまたは共用グローバル変数、すべてのローカル・グローバル変数、すべての共用グローバル変数、またはすべてのローカルおよび共用グローバル変数を除去できます。「**1 つのローカル**」 または「**1 つの共用**」オプションのいずれかで、除去するグローバル変数の名前を入力するか、またはそれに対応するドロップダウン・リストから名前を選択します。

注: 同じ .ear ファイルで稼働する別の HATS Web アプリケーションで作成された共用グローバル変数名、または同じ リッチ・クライアント環境で稼働する別の HATS リッチ・クライアント・アプリケーションで作成された共用グローバル変数名は、共用グローバル変数のドロップダウン・リストに表示されない可能

性があります。これらの共用グローバル変数の 1 つを除去するには、「**1 つの共用**」オプションの入力フィールドにその名前を入力します。

グローバル変数の詳細については、 399 ページの『第 12 章 グローバル変数との対話』を参照してください。

「グローバル変数を送信」アクション

このアクションは、HATS 標準 (JSR 168 または JSR 286) ポートレット・プロジェクトにのみ適用されます。これは、ポートレット通信ソリューションを開発する手助けとして使用できます。詳しくは、 462 ページの『ポートレット通信』を参照してください。双方向言語サポートを使用する場合の考慮事項については、 531 ページの『ポートレット・サポート』を参照してください。

JSR 168 ポートレット:

このアクションを使用して、HATS JSR 168 ポートレットからポータル・サーバー上の他の JSR 168 ポートレット (HATS または非 HATS) へ、プロパティーの java.lang.String 値としてグローバル変数の値を送信します。「プロパティー」フィールドに、送信するプロパティーの名前を入力します。名前は、受信ポートレットで使用されるプロパティー名に一致する必要があります。例えば、HATS JSR 168 受信ポートレットの「**その他**」タブの「**ポートレット**」設定で定義されているプロパティー名と同じにします。グローバル変数フィールドに、グローバル変数の名前を入力します。グローバル変数の値に索引が付いている場合は、「**拡張**」をクリックします。その後、変数の「**すべての索引を送信**」し、任意にストリング区切り文字で区切るか、または送信する「**索引**」を含めて「**単一の索引を送信**」するかを指定します。グローバル変数が共用グローバル変数である場合は、「**共用**」チェック・ボックスを選択します。

「**完了**」をクリックすると、アクションは画面のイベント (.evnt) ファイルに追加されます。また、Web サービス記述言語 (WSDL) ファイルがない場合は作成され、送信プロパティーは WSDL ファイルで定義されます。WSDL ファイルの名前は <project_name>.wsdl です。 WSDL ファイルは、WebSphere Portal ワイヤリング・ツールを使用してポートレットをワイヤリングするために必要です。 portlet.xml ファイルは、WSDL ファイルの場所を参照するように更新されます。

「**除去**」をクリックすると、アクションが画面のイベント・ファイルから除去されますが、プロパティーは WSDL ファイルから除去されません。WSDL ファイルを編集し、WSDL エディターを使用してプロパティーを除去できます。

注: WSDL ファイルからプロパティーを除去しなければ、HATS ポートレットは現状のまま正常に稼働します。

JSR 286 ポートレット:

このアクションを使用して、HATS JSR 286 ポートレットからポータル・サーバー上の他の JSR 286 ポートレット (HATS または非 HATS) へ、イベントでグローバル変数の値を送信します。「イベント名」フィールドに、送信するイベントの名前を入力します。 名前は、受信ポートレットで使用されるイベント名に一致する必要があります。例えば、HATS JSR 286 受信ポートレットの「**その他**」タブの「**ポー**

トレット」設定で定義されているイベント名と同じにします。グローバル変数フィールドに、グローバル変数の名前を入力します。

注: 単一ポートレット内のイベント名は固有でなければなりません。

必要な場合、「**拡張**」をクリックしてさらにオプションを設定します。

「詳細オプション」ページで、グローバル変数の値に索引が付いている場合は、「**変数は索引付き**」を選択します。次に、変数の「**すべての索引を送信**」するか、または送信する「**索引**」と、グローバル変数に含めるオブジェクトの「**タイプ**」を含めて、「**特定の索引を送信 (Send a specific index)**」するかを指定します。「**参照**」ボタンを使用して、有効な値タイプを入力します。選択するタイプは、順序付け可能である必要があります。順序付け可能でない場合、エラー・メッセージが表示されます。タイプは、受信ポートレットで使用されるタイプに一致する必要があります。例えば、HATS JSR 286 受信ポートレットの「**その他**」タブの「**ポートレット**」設定で定義されているタイプと同じにします。グローバル変数が共用グローバル変数である場合は、「**共用**」チェック・ボックスを選択します。

「**完了**」をクリックすると、アクションが画面のイベント・ファイルに追加され、portlet.xml ファイルが更新されます。JSR 286 ポートレット通信に関与する WSDL ファイルはありません。

一般的な考慮事項:

- このアクションはイベントの「変換を適用」アクションに従う必要があります。
- 以下のアクションは、「変換を適用」アクションと「グローバル変数を送信」アクション間でサポートされる唯一のアクションです。
  - ビジネス・ロジックを実行
  - グローバル変数を抽出
  - データを挿入
  - グローバル変数を設定
  - グローバル変数を除去
  - マクロ・トランザクションを実行
  - 一時停止
- 上記のアクションは、他のポートレットに送信する前にグローバル変数を処理する場合に便利です。 例えば、ログイン画面で入力したユーザー ID を他のポートレットに送信する場合などです。この例では、ログイン画面を変換するために「変換を適用」アクションを追加し、ユーザー ID 入力フィールドの領域を抽出するために「グローバル変数を抽出」アクションを追加して、それをグローバル変数に保存します。その後、グローバル変数に保管されたユーザー ID を他のポートレットに送信するために、「グローバル変数を送信」アクションを追加します。
- 以下のアクションは、「変換を適用」アクションと「グローバル変数を送信」アクション間でサポートされていません。これらのアクションのいずれかが「変換を適用」アクションと「グローバル変数を送信」アクション間に配置されている場合、通常どおりに実行されますが、グローバル変数は送信されません。
  - 変換を適用
  - URL を表示

- URL に転送
- マクロを実行
- キーを送信
- 切断
- ポートレットのプレプレゼンテーション (ブロック区切り文字)

• グローバル変数に含まれるオブジェクトの「**タイプ**」指定は、JSR 286 ポートレットの場合にのみサポートされます。JSR 168 プロジェクトの画面イベント・ソースにタイプ・パラメーターを手動で追加すると、それは無視されます。

「URL または SWT コンポジットを表示」アクション RCP のみ

このアクションは、「変換」ビューの変換領域に Web ページまたは SWT コンポジットを表示する場合に使用します。Web ページを表示する場合は、Web ページの URL (Uniform Resource Locator) アドレスを「**URL**」フィールドに指定します。SWT コンポジットを表示する場合は、そのクラス名を「**コンポジット・クラス**」フィールドに入力するか、「**参照**」ボタンを使用して、リッチ・クライアント・プロジェクト内から SWT コンポジットを選択します。 Web ページまたは SWT コンポジットは、テンプレートのリストから選択したテンプレートに囲まれて表示されます。 Web ページまたは SWT コンポジットが読み込まれたら、ユーザーはいずれかの「**続行**」ボタンをクリックして、HATS アプリケーションに戻る必要があります。

注: 「**続行**」ボタンを含む SWT コンポジットの実装方法については、「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」 を参照してください。

「URL を表示」アクション Web のみ

Web ページをこの画面イベントのアクションとして表示する場合、「**URL**」フィールドに Web ページの URL (Uniform Resource Locator) アドレスを指定する必要があります。Web ページは、変換の表示と同様にデフォルト・テンプレートで囲まれて表示されます。Web ページが読み込まれた後は、Web ページの下部にある「**続行**」ボタンをクリックすることにより、ユーザーは HATS アプリケーションに戻ります。

組み込みオブジェクト (例えば HTML のデータ・タイプを持つオブジェクト) が、ブラウザーによってサポートまたは許可されていない場合、このアクションは Web ページの内容の代わりに、Web ページへのリンクを表示します。サポートされる Web ブラウザーのリストおよび制限については、「Rational Host Access Transformation Services: System Requirements」(http://www.ibm.com/support/docview.wss?uid=swg27011794) および「Host Access Transformation Services 9.5 - Known issues and workarounds」(http://www.ibm.com/support/docview.wss?uid=swg27046998) を参照してください。

「URL を表示」アクションのターゲットを指定する場合は、クライアントのワークステーションからターゲット・ページにアクセスできる必要があります。そのため、クライアントのワークステーションにターゲット・ページへの直接アクセス権限がない場合などは、企業のファイアウォールで保護されているターゲット Web ページは指定できません。ターゲットの完全な URL を指定してください。そうしないと、クライアントはターゲット・ページのホスト名を解決できません。HATS

アプリケーションをホストする (「クロスドメイン」アクセス) アプリケーション・サーバーにターゲット・ページが存在しない場合、サーバーにないターゲット・ページをロードできるようにブラウザー設定を調整する必要があります。

例えば、Internet Explorer V6.0 以降で、「**ツール**」メニューに進み、「**インターネット オプション**」を選択して、「**セキュリティ**」タブを選択します。ブラウザー・ウィンドウの右下角に Web コンテンツ・ゾーンが表示されます。このページがインターネット・ゾーンにある場合、ターゲット・ページがロードできるように、インターネット・ゾーン設定を調整する必要が生じることがあります。正しいゾーンを選択し、「**レベルのカスタマイズ**」ボタンをクリックします。「**各種**」の設定にスクロールして、「**ドメイン間でのデータ ソースのアクセス**」の設定を選択します。ターゲット・ページをロードするには、「**使用可能**」を選択します。ターゲット・ページのロード前にプロンプトを出すには、「**プロンプト**」を選択します。「**無効にする**」を選択すると、ターゲット Web ページがロードされません。

「URL に転送」アクション Web のみ

「URL に転送」アクションにより、変換指向 HATS Web アプリケーションから 1 つ以上のチェーン統合オブジェクトを起動する JSP に制御を渡すことができます。これにより、作成済みの統合オブジェクトを使用できます。統合オブジェクトは既存の接続またはバックグラウンド接続を使用できます。

このアクションは、HATS Web プロジェクトおよび HATS 標準ポートレット・プロジェクトでサポートされています。

転送アクションをイベントに追加するには、以下のパラメーターを指定する必要があります。

- コントロールを渡す JSP。統合オブジェクトの呼び出しに加え、情報が必要な場合は、この JSP はユーザーと対話できます。

  注: コントロールを渡す JSP が HATS 標準ポートレット・プロジェクトで生成されたモデル 1 JSP である場合は、以下のステートメントが JSP に追加されなければなりません。

  ```
  <%@ taglib uri="http://java.sun.com/portlet" prefix="portletAPI" %>
  <portletAPI:defineObjects/>
  ```

- 実行する最初の統合オブジェクトの startStateLabel。デフォルト接続を格納して、その接続が統合オブジェクトに対して確実に使用可能になるように、この値は HATS アプリケーションによって使用されます。

  注: このパラメーターは、デフォルト接続を JSP/統合オブジェクト・ロジックに渡す必要がある場合にのみ必要です。

注: 大規模な JSP を実行する場合は、JVM システムのプロパティー **com.sun.tools.javac.main.largebranch=true** を設定し、サーバーを再始動します。JVM のプロパティーを設定するには、管理コンソールを使用するか、JVM の引数を設定するか、またはカスタム・プロパティーを作成します。

URL への転送により、JSP/統合オブジェクト・ロジックが使用する接続を複数の方法で管理できます。例えば、次のようになります。

- HATS アプリケーションはすでにデフォルト接続を確立しており、JSP/統合オブジェクト・ロジックでこの接続を使用するとします。この場合、制御を受け取る

JSP は、チェーンの先頭ではない統合オブジェクトを駆動する必要があります。駆動される統合オブジェクトは、作成時に統合オブジェクトに関連付けられた開始状態ラベルを使用して HTTP セッション・オブジェクトで接続を検索することによって、使用する接続を探します。このラベルは、「URL に転送」の定義の一部として指定されたラベルに一致する必要があります。この場合、制御が変換指向のプロジェクトに戻されると、ホスト画面が更新されたかどうかがチェックされます。変更された場合は、現在のイベントに関連付けられたアクション・リストにある残りのアクションはすべて無視されます。ホスト画面が変更されなかった場合は、残りのアクションは無視されません。

- HATS アプリケーションは既にデフォルト接続を確立しているが、JSP/統合オブジェクト・ロジックで新規バックグラウンド接続を使用するとします。この場合、制御を受け取る JSP は、チェーンの先頭であるかまったくチェーニングされていない統合オブジェクトを駆動する必要があります。この統合オブジェクトにより、新規バックグラウンド接続が確立され、その接続に対して指定のタスクが実行されます。この場合は「URL に転送」の定義で startStateLabel を指定する必要はありません。
- デフォルト接続が開始されていない場合。これは、開始イベントまたは接続イベント (取得アクションを実行する前) に「URL に転送」アクションを追加する場合のケースです。このシナリオでは、多くの可能性があります。例えば、制御を受け取る JSP が、専用のバックグラウンド接続を取得し、指定のタスクを実行し、HATS アプリケーションに制御を戻すようにプログラミングされた統合オブジェクトを駆動できます。この時点で、HATS アプリケーションがイベント処理を続行し、デフォルト接続を確立できます。別の可能性として、JSP がチェーンの先頭である統合オブジェクトを駆動する場合も考えられます。統合オブジェクトが接続を確立し、タスクを実行し、デフォルト接続として使用する接続を HATS アプリケーションに渡します。
- 接続が HATS 標準ポートレットによって確立される際、その接続は、固有の接続 ID および「URL に転送」アクションの開始状態ラベルから構成されたキーを使用して保存されます。この同じキーは、制御を取得する最初の統合オブジェクトによって使用される必要があります。キーは要求オブジェクトに、属性 CommonConstants.HPUB_LINK_KEY として保存されます。要求オブジェクトからリンク・キーを取り出し、統合オブジェクトに setHpubLinkKey() メソッドを呼び出すために、JSP を編集してから、IntegrationObject processRequest() メソッドを呼び出してください。例えば、次のようになります。

```
ExampleIO.setHPubLinkKey
  ((String)request.getAttribute(CommonConstants.HPUB_LINK_KEY));
```

接続がチェーンの最初の統合オブジェクトによって確立されると、チェーンの最初の統合オブジェクトによって作成されたリンクを使用して保存されます。

両方のシナリオで、後続の統合オブジェクトは同じリンクを再使用する必要があります。そのリンクは getHPubLinkKey() メソッドを呼び出す最初の統合オブジェクトから取り出すことができます。必要に応じて、リンクを要求オブジェクトのパラメーターとして、後続の JSP に受け渡すことができます。これは非表示入力パラメーターを、以下に示す形式に追加することによって実行できます。

```
<INPUT NAME="<%= CommonConstants.HPUB_LINK_KEY %>"
       VALUE="<%= ExampleIO.getHPubLinkKey() %> "TYPE="hidden">
```

後続の JSP は、processRequest() メソッドを呼び出す前にキーを取り出しそれを IO に設定するために、以下のステートメントを使用します。

```
ExampleIO_2.setHPubLinkKey
 ((String)request.getParameter(CommonConstants.HPUB_LINK_KEY));
```

転送アクションを使用する場合は、統合オブジェクトの実行後に制御が自動的に HATS アプリケーションに戻されることはありません。JSP は、HATS アプリケーションに明示的に制御を戻す必要があります。転送アクションより前にデフォルト接続が確立されていない場合に統合オブジェクトがデフォルト接続を使用する場合は、デフォルト接続として使用する接続を HATS アプリケーションに渡す必要があります。この場合、必ず HttpServletRequest で要求パラメーターを設定してから、その要求を HATS アプリケーションに転送してください。パラメーターは CommonConstants.HATS_EXISTING_CONN です。このパラメーターに必要な値を取得するには、チェーンの最後の統合オブジェクトで getHPubEndChainName メソッドを呼び出します。

例えば Web プロジェクトでは FORM は以下のようになる可能性があります。

```
<FORM NAME="exampleLink" METHOD="GET"
ACTION='<%= response.encodeURL(request.getContextPath()+"/entry")%>'>
<INPUT TYPE="HIDDEN" NAME="<%= CommonConstants.HATS_EXISTING_CONN %>"
VALUE="<%= ExampleIO.getHPubEndChainName()%>" />
<INPUT TYPE="submit" VALUE="Return to HATS application" />
</FORM>
```

また、「**HATS ツール**」メニューを使用して「**HATS アプリケーションへの転送の挿入**」を行うこともできます。このオプションは、1 つの例外を持つコードを自動的に挿入します。以下の行は含まれません。

```
<INPUT TYPE="HIDDEN" NAME="<%= CommonConstants.HATS_EXISTING_CONN %>"
       VALUE="<%= IntegrationObjectName.getHPubEndChainName() %>" />
```

HATS アプリケーションに接続を渡す場合は、JSP を編集して、「FORM NAME=」コード行の直後にコード行を追加する必要があります。

その他の場合 (デフォルト接続をオープンしてから転送アクションを起動する場合や、統合オブジェクトが転送アクションを使用しない場合) は、接続を戻す必要はありません。これらの場合、記載の例から最初の INPUT ステートメントを省略できます。

注: JSP/統合オブジェクト・ロジックを HATS アプリケーションへの最初の入り口にすることができます。この場合、HATS アプリケーションを駆動し、JSP/統合オブジェクト・ロジックで確立した接続を HATS アプリケーションに渡し、デフォルト接続として使用させるように指定したコードを使用します (接続をデフォルト接続定義から作成した場合)。

マクロを実行

マクロを記録した場合は、そのマクロをドロップダウン・ボックスから選択して実行できます。このマクロは、このイベントが選択されたときに実行を開始します。「**マクロを実行**」アクションは現行接続でマクロを実行することに注意してください。これに対し、「**マクロ・トランザクションを実行**」は新規接続を作成してその接続でマクロを実行します。

マクロを実行するには、「**アクションの追加**」リストから「**マクロを実行**」項目を選択し、実行するマクロの名前をドロップダウン・リストから選択します。実行するマクロをこの画面イベントのアクションとして定義した場合は、そのマクロは最後に適用されます。対象の画面を認識するために、必ずきわめて明確な基準を使用し、他の画面ではマクロが実行されないようにします。例えば、ストリング基準を使用する場合は、画面の任意の場所ではなく、正確な場所で検出されなければならないことを指定してください。マクロが終了する画面が、マクロを開始する基準を満たしてはなりません。マクロを記録する場合は、画面イベントによって認識される画面が最終画面にならないようにしてください。認識された画面がマクロの最終画面である場合、ループが作成され、マクロ実行の実行時に画面の強制変更するキーを送信するオプションは実行されません。

通常、変換を適用した後に同じ画面でアクションとしてマクロを実行することはありません。両方のアクションを指定すると、変換が適用され、ユーザーが変換された画面と対話した後のみにマクロが実行されます。マクロの実行は、常に最後のアクションとなります。画面が認識された場合にマクロを実行する場合は、その画面に変換を適用しないでください。その画面に画面イベントを作成する場合は、必ずウィザードの「アクション」ページにある「**変換を適用**」チェック・ボックスをクリアしてください。マクロがナビゲートする最後のホスト画面で使用する別の画面イベントを作成し、変換を適用して最終画面から派生する Web ページの外観を制御できます。

変換にマクロ・ボタンを挿入して、ユーザーが変換された画面からマクロを実行できるようできます。この場合ユーザーは、変換された画面と対話することも、マクロ・ボタンをクリックして現在のホスト画面でマクロの稼働を開始することもできます。

マクロは、ホスト端末を使用して HATS Toolkit に記録できます。マクロのインポートの詳細については、381 ページの『第 11 章 マクロとホスト端末』を参照してください。

マクロ・トランザクションを実行

このアクションを選択すると、指定した接続 (別のホストへ接続でも可) の新規インスタンスの記録済みのマクロを実行できます。「**マクロ・トランザクションを実行**」 ボタンをクリックした後、「**マクロを実行**」ドロップダウン・リストから実行するマクロを選択し、使用する接続を「**接続時に実行**」ドロップダウン・リストから選択します。

注: 指定した接続に接続マクロが定義されている場合、選択したマクロの実行前に、その接続マクロが実行されます。

キーを送信

このアクションは、ホスト画面に指定のキーを送信します。画面上でアクションを適用する場所を選択したら、ホスト画面に送信するキーを選択します。

黒丸付きリストから「**PF および PA キー**」 または「**他のホスト・キー**」のいずれかを選択します。その後、選択した項目のドロップダウン・メニューをクリックしてキーを選択します。

選択したキーによってホスト画面が変化したり、アクションが無限ループに入ったりする場合があることに注意してください。

アクション・リストの順序を考慮する場合は、必ず「**キーを送信**」アクションを最後に実行してください。

切断

「**切断**」アクションでは、すぐに切断イベントが実行されます。デフォルトでは、切断イベントによって、デフォルトの接続が切断および解放されます。

一時停止

「**一時停止**」アクションでは、通常処理を続行するまでの時間 (ミリ秒単位) を指定できます。

ポートレットのプレプレゼンテーション

「**ポートレットのプレプレゼンテーション (Portlet Prepresentation)**」を使用する場合は、「**アクション**」タブの「**ブロック区切り文字を追加**」ボタンを使用して、「**プレプレゼンテーション開始**」区切り文字と「**プレプレゼンテーション終了 (End)**」区切り文字を追加できます。

このアクションは、HATS 標準ポートレットではサポートされていません。

「**プレプレゼンテーション開始**」と 「**プレプレゼンテーション終了 (End)**」を使用して、ポートレット処理のプレプレゼンテーション・フェーズで実行する必要のあるアクションを囲むことができます。ポートレットのメッセージングまたは連携ポートレットをインプリメントする場合は、「**ブロック区切り文字を追加**」ボタンを使用します。プレプレゼンテーション開始区切り文字を挿入するには「**ブロック区切り文字を追加**」ボタンを一度クリックします。プレプレゼンテーション終了区切り文字を挿入するには、「**ブロック区切り文字を追加**」を再度クリックします。ブロック区切り文字で囲まれた、実行する必要があるアクションを、リスト内で上下に移動するには、「**上へ**」ボタンと「**下へ**」ボタンを使用します。アクションを起動するタイミングに応じて、プレプレゼンテーション・ブロックの場所が決定されます。画面イベントを実行する場合、最初に、または「**アクション**」リストの先頭でプレプレゼンテーション・アクションを起動できます。画面でユーザーが対話した後にプレプレゼンテーション・アクションを実行する場合、「**アクション**」リストの下部でイベントを起動します。ポートレットでのブロック区切り文字の使用方法について詳しくは、*「HATS Web アプリケーション・プログラマーズ・ガイド」*を参照してください。

グローバル規則

グローバル規則を使用すると、ホスト入力フィールドのパターン認識と変換が可能になります。また、グローバル規則では、カスタマイズした画面、およびカスタマイズしていない (デフォルト・レンダリングの) 画面が処理されます。グローバル規則は、プロジェクト・レベルと、画面レベルの両方で定義できます。このタブは、画面レベルのグローバル規則を指定するときに使用します。グローバル規則を定義する方法の詳細については、 116 ページの『グローバル規則』を参照してください。

画面レベルのグローバル規則を適用すると、画面に対してデフォルトのレンダリングを使用できるだけでなく、画面上の入力フィールドをカスタマイズできます。例えば、画面レベルのグローバル規則で、「**指定した領域内で入力フィールドを検索**」パターン・タイプを使用すると、ある画面上でカスタマイズする入力フィールドを正確に特定する (例えば、ある特定のテーブル内のデータ) ことができる一方で、その画面に対して引き続きデフォルト・レンダリングを使用できます。

プロジェクト・レベルと画面レベルの両方のグローバル規則が同じ入力フィールドに定義されている場合は、画面レベルの規則が優先されます。

テキスト置換

「**テキスト置換**」タブにはテーブルが表示されますが、ここでは、置換の対象にする元のホスト画面の保護テキストと、置き換えとして使用するテキスト、HTML コンテンツ Web のみ、またはイメージを合わせて指定できます。 さらに、テキスト検索で大文字と小文字を区別するかどうか、正規表現サポートを使用するかどうかも表示されます。「**追加**」ボタンをクリックして、テキスト置換パラメーターを新規作成します。

選択ウィンドウから置換するテキストを選択するか、「**置換**」テキスト・ボックスにテキストを入力します。

「**大文字と小文字を区別**」チェック・ボックスを選択すると、「**置換**」テキスト・ボックスに入力したテキストに完全に一致するもののみが検索されます。

テキストで置換するには、置換されるテキストを下のウィンドウに入力します。

HTML コード Web のみ を入力すると、コードは HATS よって赤で強調表示されるため、コードが有効かどうかが分かります。

「**挿入**」アイコンをクリックすると、ボタンやリンクを追加できます。「**ボタンの挿入**」と「**リンクの挿入**」の設定は、以下のとおりです。

表題　レンダリングされたボタンやリンクに表示するテキスト。

アクション・キー
ボタンやリンクがクリックされたときに送信するホスト AID キー。

スタイル・クラス Web のみ
ボタンやリンクの外観を制御する Cascading Stylesheet (CSS) クラス名。ボタンのデフォルト値は、HATSBUTTON です。リンクのデフォルト値は、HATSLINK です。

また、「**イメージ**」ラジオ・ボタンをクリックしてドロップダウン・リストからイメージを選択するか、「**インポート**」 ボタンをクリックすることによって、テキストをイメージで置換することもできます。

「**正規表現**」チェック・ボックスを選択すると、Java の正規表現がテキスト置換アルゴリズムの一部としてサポートされます。正規表現は、ストリング・セットを表す文字のパターンです。正規表現は、パターンの出現の検索と変更に使用できます。例えば、以下のように指定します。

```
Replace: ([¥.¥-|_¥w]+)@([¥-¥_¥.¥w]+)
With: <a href="mailto:$1@$2" class="HATSLINK">$1@$2</a>
```

この結果、以下のように置換されるため、ホスト画面上の E メール・アドレスがリンクに変換されます。

```
Original host screen text: user@company.com
Replacement text:
<a href="mailto:user@company.com" class="HATSLINK">user@company.com</a>
```

値のテーブルの右側にあるボタンを使用して、テキスト置換の仕様を追加、変更、または除去できます。テキストは、プロジェクト・レベル、レンダリング項目レベル、変換レベル、およびコンポーネント・レベルで置換できます。

テキスト置換リストを使用する場合、1 回のテキスト置換によって作成されたテキストまたは HTML が、後続のテキスト置換によってさらに置換されることがあります。例えば、ホスト画面上のあるストリングを `<img src="yourImage.jpg">` で置換し、後続のテキスト置換で `sr` を別のストリングで置換すると、`src` パラメーターが変更されているため、`<img>` タグは正常に機能しなくなります。この状態を回避するには、テキスト置換項目の順序を変更するか、より具体的な指定を行って、前のストリングが誤って置換されないようにします。

注:

1. テキスト置換を使用するときは、注意が必要です。ストリングの文字数が異なるテキスト置換を行うと、画面の GUI 表示が変わってしまうことがあります。画面領域の表示に使用されるウィジェットによって、画面の 1 行に収められているテキストが縮小または拡張されたり、次の行に強制改行されたりすることがあります。また、通常は、1 文字しかないストリング、保護フィールドが間に存在するテキストで構成されるストリングなどを置換することは、非効率的です。
2. デフォルトでは、テキスト置換により、置換前のテキストより長いストリングが切り捨てられます。変換で、元のホスト画面のレイアウトを維持するために、テキストが切り捨てられることがあります。切り捨ては、フィールド・ウィジェット (場合によってはテーブル・ウィジェット) を使用してレンダリングされたテキストにのみ発生します。

   ソース設定の **truncateToPreserveLength** を追加することにより、短ストリングを長ストリングで置換するテキスト置換を使用可能にすることができます。この設定の値は、`true` および `false` です。設定を指定しなかった場合のデフォルト値は `true` で、プロジェクトに対する影響はありません。`false` を指定すると、長いテキスト置換が使用可能になります。この設定を使用する場合は、デフォルト・レンダリングを使用してはなりません。デフォルト・レンダリングを使用すると、設定が無視されます。

   truncateToPreserveLength 設定は、プロジェクト・レベル設定です。プロジェクト内のテキスト置換すべてに作用します。この設定を使用するときは、すべてのレンダリングが思ったとおりに表示されること、および位置合わせの問題が発生したとしてもそれが問題でないことを確認してください。

   プロジェクト全体の設定を有効にするには、application.hap ファイルのソース・ビューを開きます。変換のクラスを見つけ、次の例に示す強調表示された設定を追加してください。

   ```
   <class name="com.ibm.hats.transform">
      <setting name="truncateToPreserveLength" value="false" />
   </class>
   ```

3. 選択テキスト・ウィンドウは便利ですが、実際には、「**置換**」フィールドに表示されるテキストが置換されます。テキスト置換を複数の行または無保護フィールドにまたがって適用することはできません。プレビュー・フィールドを使用して、複数のテキスト行を選択および置換することはできません。
4. プロキシー・サーバーを通じて HATS アプリケーションがアクセスされたときにテキストをイメージで置き換えるには、プロキシー・サーバーを使用するように HATS アプリケーションを構成する必要があります。手順については、44 ページの『プロキシー・サーバーを使用するための HATS アプリケーションの構成』を参照してください。
5. テキストとイメージの置換の使用と、リッチ・クライアント・アプリケーションでの特定の SWT ウィジェットの使用とを組み合わせた場合には、これに関連した制限事項が存在します。詳しくは、103 ページの『HATS リッチ・クライアントの考慮事項と制限』を参照してください。

次の画面

「**次の画面**」タブでは、現在表示中の画面の後に表示する次の画面を指定できます。

「**追加**」ボタンをクリックして、リストに画面を追加します。**上矢印**ボタンまたは**下矢印**ボタンを使用して画面の比較順序を変更したり、「**除去**」ボタンを使用して画面を除去したりできます。

リスト内で次に表示する画面が一致しない場合、アプリケーションで「**アプリケーション・イベント優先順位リストの検索 (Search application event priority list)**」または「**アプリケーション・イベントの実行 (Execute application event)**」を実行できます。

「**検索アプリケーション・イベント優先順位リスト**」オプションでは、次の画面のプロジェクト設定でイベント優先順位リストが検索されます。これはデフォルトです。「**アプリケーション・イベントの実行 (Execute application event)**」ドロップダウン・リストでは、リスト内の次の表示画面が一致しない場合に発生させるエラーなどのイベントを選択できます。イベント・リストは、以下のとおりです。

- 一致しない画面
- エラー
- 切断
- 停止

詳しくは、125 ページの『アプリケーション・イベント』を参照してください。

次に表示される画面を予測できる場合は、「**次の画面**」を使用するとパフォーマンスが向上します。

ソース

「**ソース**」タブでは、*se-name*.evnt ファイル内のタグと値が表示されます。これらは画面イベント用に提供されているすべての情報です。ここで、*se-name* は、画面イベントを作成したときに指定した名前です。画面イベント・エディターのほかのタブを変更すると、「**ソース**」タブに表示されるタグと値もそれに合わせて変更されます。

ソース・ファイル内のタグと値を変更することもできます。その変更は、画面イベント・エディターの適切なタブに反映されます。

使用禁止画面

HATS 接続のオペレーター情報域 (OIA) 状況は、画面が構成済み画面イベントの 1 つに一致しているかどうかを判断するために HATS が使用する画面マッチング基準の 1 つです。 OIA 状況が一致しない場合、HATS は対応する画面イベントとは一致せず、カスタム変換またはアクションではなくデフォルトの変換が表示されます。このセクションでは、入力禁止 (キーボードがロックされている) 画面または OIA 状況が不一致の画面を処理する際に使用できる方法について説明します。

認識基準

デフォルトでは、HATS によって、OIA 状況に関する以下の認識基準がすべての画面イベントに追加されます。

```
<oia invertmatch="false" optional="false" status="NOTINHIBITED"/>
```

これは、他のマッチング基準に加えて、OIA 状況は HATS が画面を一致していると評価する NOTINHIBITED でなければならないことを意味します。 OIA 状況が他のものであれば、他の基準が満たされていても、画面は一致していると見なされません。結果として、デフォルトの変換がブラウザーに表示されます。

特定の画面イベントに関して HATS が OIA 状況を無視するようにするには、以下の手順に従って、認識基準を単に NOTINHIBITED から DONTCARE に変更してください。

1. 「HATS Projects」ビューで、ご使用のプロジェクト、および画面イベント (画面カスタマイズまたは画面組み合わせ) のリストを展開します。
2. 画面イベントをダブルクリックして、画面イベント・エディターを開始します。
3. エディターの下部にある「**ソース**」タブをクリックします。
4. ストリング status="NOTINHIBITED" を含む行を検索します。NOTINHIBITED を DONTCARE に変更します。
5. 「**ファイル**」>「**保存**」をクリックするか、または「**Ctrl+S**」を押すことで保存します。

ホスト画面が実際に使用禁止になっているかロックされている場合、Ctrl+R (ホストのリセット) またはエスケープ (ホストのクリア) などの特定のキーまたはキー組み合わせを押して、画面をアンロックするかまたは次の画面に移動してください。

使用禁止画面の自動処理

使用禁止またはロックされている画面を HATS が自動的に認識し処理できるように、そのための特別な画面カスタマイズを作成して、これを実現できます。このためには、以下の手順に従います。

1. 画面カスタマイズの「画面認識基準」タブで、すべての基準をクリアし除去します。
2. 画面カスタマイズの「ソース」タブで、invertmatch="true" を持つように OIA 行を変更します。これを行うと、以下のようになります。

```
<oia invertmatch="true" optional="false" status="NOTINHIBITED"/>
```

3. 画面カスタマイズの「アクション」タブで、OIA 状況をクリアして NOTINHIBITED に復元するアクションを追加します。状況によって、取るべき正しいアクションは異なります。例えば、OIA 状況をクリアして NOTINHIBITED に復元するには、単にホスト・キー Enter (または他のホスト・キー) を送信するだけ十分な場合があります。しかし、他の場合では、ホスト接続を OIA 状況が NOTINHIBITED であった状況に戻すために、カーソル位置を変更するか、一連のホスト・キーを送信するマクロを実行する必要がある可能性があります。
4. 「HATS プロジェクト設定」エディターで、「**イベント**」タブを選択し、特別な画面カスタマイズがリストの先頭に来るように「画面イベントの優先順位」リストをソートします。これによって、他の画面イベントとのマッチングを試行する前に、HATS は最初に INHIBITED OIA 状況を使用して画面を検出および処理するようになります。

複数の使用禁止画面の処理

場合によっては、複数の使用禁止またはロックされた画面があり、その条件をクリアするにはそれぞれ異なるアクションが必要である可能性があります。そのため、そのようなケースを処理するために複数の特別な画面カスタマイズを作成し使用する必要がある可能性があります。「ソース」タブの OIA 行の invertmatch を true に変更するのに加えて、「画面認識基準」タブに他の基準を追加し、各画面カスタマイズが固有で、該当するときにのみ起動されるようにする必要がある場合があります。

例えば、ユーザーが保護フィールドにテキストを入力しようとすると、ホスト画面がロックされる可能性があります。この場合はおそらく、表示の保護領域上にカーソルがありますというメッセージが表示されます。この条件をクリアするには、ユーザーは無保護領域にカーソルを移動しなければなりません。これを自動化するために、特別な画面カスタマイズを作成できます。OIA 領域に invertmatch="true" を設定するのに加えて、メッセージ・テキスト 表示の保護領域上にカーソルがあります の検索も行います。一致するものが存在すれば、そのカスタマイズは、カーソルを保護領域から移動し OIA 状況をクリアするマクロを実行するアクションを持つことができます。

同じシステム上で、例えばユーザーが無効なファンクション・キーを押した場合に画面がロックされるという、2 つ目の条件があることがあります。この場合、エラー画面上に 無効なファンクション・キーが押されました のテキストが表示されます。この条件をクリアするには、ユーザーは Enter キーを押す必要がある可能性があります。これを自動化するために、別の特別な画面カスタマイズを作成できます。ここでも OIA 領域に invertmatch="true" を設定しますが、テキスト 無効なファンクション・キーが押されました と突き合わせます。このアクションのため、この画面カスタマイズは Enter キーをホストに送信します。

特定の環境によっては、単に OIA 基準を DONTCARE に設定し、端末エミュレーターを使用して処理するように、ユーザーがその状態を処理できるようにすることによって、使用禁止画面を処理することが、最も容易である場合があります。しかし上記で示したように、1 つ以上の特別な画面カスタマイズを作成すれば、HATS が使用禁止画面を自動的に認識し処理できるようにすることも可能です。

BMS マップ・セットのインポート

HATS では、顧客情報管理システム (CICS®) の基本マッピング・サポート (BMS) マップ・セットから画面キャプチャーを作成できます。BMS マップは、CICS の画面定義ファイルです。各マップでは、画面のすべてまたは一部を定義します。CICS アプリケーションは、通常、1 つ以上のマップを表示して完全な画面イメージを作成します。マップには、静的または動的な区域またはフィールドの両方が含まれます。

BMS マップのソースは、マップ・セットというグループに編成されます。1 つのマップ・セットには、1 つ以上のマップが含まれています。マップ・セットは、ソース形式で存在します。各ソース・ファイルに 1 つのマップ・セットが含まれます。HATS で BMS マップをインポートする場合、マップ・セット・レベルでインポートが発生します。個別のマップをインポートすることはできません。

マップは、最初、ホスト・システムに存在するので、ご使用のローカル・ファイル・システムに配置する (またはローカル・ファイル・システムに表示されるようにする) 必要があります。HATS にインポートする前のマップの拡張子は、.bms です。

まず、ホストからローカル・ファイル・システムに BMS マップ・セットをコピーします。この操作は、HATS の外部で手動で行います。具体的には、ファイルをファイル・システムに物理的にコピーするか、またはネットワーキング・ソフトウェアを使用してファイル・システムをホストに接続します。ファイルは、バイナリー・ファイルとして転送しないでください。ファイルは、ワークステーションでローカルに読み取ることができる必要があります。

1. 「**ファイル**」>「**インポート**」>「**HATS**」>「**HATS への BMS マップ・セット**」をクリックして、「インポート」ウィザードを開きます。
2. リストからインポートする BMS マップ・セット・ファイルとインポートするファイルの宛先を選択します。
3. プルダウン矢印をクリックして、国別の「**ホスト・コード・ページ**」を選択します。 これは、ホストで BMS マップを作成したときに使用したコード・ページです。
4. プルダウン矢印をクリックし、「**BMS ファイル・コード・ページ**」を選択します。これは、BMS ファイルをワークステーションに転送するときに使用したコード・ページです。
5. 「**警告せずに既存のマップを上書き (Overwrite existing maps without warning)**」オプション、および「**マップ名を画面キャプチャー名として使用し、マップごとに 1 つの画面キャプチャーを生成**」オプションを使用できます。

   注: マップ同士を組み合わせて画面を作成する必要がある場合、画面キャプチャーを自動的に作成しない方がよい場合があります。
6. 「**警告せずに既存の画面キャプチャーを上書き**」を選択することもできます。
7. 「**完了 (Finish)**」をクリックする。

BMS キャプチャー・ファイルは、HATS プロジェクト内の独立した **Maps** フォルダーに保管されます。デフォルトでは、インポートされたマップがない場合、「HATS Projects」ビューに **Maps** フォルダーは表示されません。**Maps** フォルダー

内には、各マップ・セットに対する個別のフォルダーが含まれます。各フォルダー内には、各マップに対する別個のファイル (拡張子は .bmc) が含まれます。また、元のソースはファイル・システムに保管され (拡張子は .bms)、関連付けられたマップを含むフォルダー内に表示されます。ソースを編集するには、ファイルをダブルクリックします。ソースを編集して保存すると、マップが自動的に再生成されて上書きされます。ただし、「**プロパティー**」ビューから変更されたマップを除きます。

「**ナビゲーター**」ビューで BMS ソース・ファイルを *<project>*/bmsmaps フォルダーにドラッグして、BMS マップをインポートすることもできます。ファイルは、サブフォルダーではなく、bmsmaps フォルダーに直接格納してください。この方法でマップをインポートしている間にエラーが発生した場合、「**問題**」ビューにメッセージが表示されます。

注: ソースは、参照としてだけ保存できます。エディターに表示できる以外、プロジェクトに機能はありません。ソースを表示すると、マップ・セット全体が表示されます。エディター内で表示されている特定のマップの定義が強調表示されることはありません。

BMS マップのインポート後には、「HATS Projects」ビューから実行できる複数のアクションがあります。

- マウス・カーソルをマップ名の上で移動すると、マップの小さいイメージが表示されます。
- 「HATS Projects」ビューでマップをクリックすると、「**プロパティー**」ビューが更新され、選択されたマップの詳細が表示されます。マップ内に名前付きフィールドがある場合、「**プロパティー**」ビューを使用して、これらのフィールドのコンテンツを変更できます。マップのコピーを作成して、そのコピー内の名前付きフィールドのコンテンツを変更すると、アプリケーションによる画面の表示方法を正確に反映した別のカスタマイズ・バージョンのマップを作成できます。
- マップをダブルクリックすると、3270 画面でのマップの表示方法を示すエディター・ページが表示されます。マップ・セット・フォルダー名をダブルクリックすると、エディターにマップ・ソースが表示されます。
- 「HATS Projects」ビューでマップ名を右クリックすると、ポップアップ・メニューに「**画面キャプチャーの生成**」選択項目が追加されます。これを選択すると、1 つ以上の BMS キャプチャー・ファイルを選択して画面キャプチャーの作成に使用するための「画面キャプチャーの生成」ウィザードが起動します。選択した各 BMS マップに対して個別の画面キャプチャーを作成したり、選択した BMS マップを単一の画面キャプチャーにマージしたりすることもできます。フィールドがオーバーラップしている場合は、マップをマージできません。必要に応じて、「**警告せずに既存のリソースを上書き**」チェック・ボックスも選択できます。

  注: 個別の画面キャプチャーが生成され、複数のファイルが選択されている場合、画面キャプチャー・ファイルの名前が自動的に生成されます。

画面キャプチャーの作成が完了したら、HATS 画面イベントの作成を開始できます。

第 8 章 変換の操作

変換とは、特定のホスト画面をカスタマイズする方法を定義する JSP ファイル (Web プロジェクトの場合) または SWT コンポジット (リッチ・クライアント・プロジェクトの場合) のことです。変換を作成するときは、変換の土台となるホスト画面を選択することができます。ホスト画面は、ホスト端末画面または画面キャプチャーを土台にすることができます。画面キャプチャーは、インポートした BMS マップ (3270 のみ) から取得することもできますし、ホスト端末から直接取得することもできます。

変換の適用は、画面イベントで指定できるアクションの 1 つです。変換は一般的に、以下のような場合に使用されます。

- ホスト画面情報の表示方法を変更する。
- ユーザーに表示したくないホスト画面情報をフィルターにかける。
- Web 表示でホスト・コンポーネントをウィジェットとして表示する。

「変換の作成」ウィザード

「変換の作成」ウィザードを使用して画面変換を定義します。ウィザードにアクセスするには、以下のようにさまざまな方法があります。

- 「**画面キャプチャー**」フォルダー内にある任意の画面キャプチャーを右クリックし、「**新規 HATS**」>「**変換**」を選択します。
- HATS ツールバーで「**HATS 変換を作成**」アイコンをクリックします。
- HATS メニュー・バーで、「**HATS**」>「**新規**」>「**変換**」 (または「**ファイル**」>「**新規**」>「**HATS 変換**」) を選択します。
- 画面カスタマイズを作成できる場所ならどこでも、変換を作成することができます。詳細については、173 ページの『第 7 章 画面イベントの処理』を参照してください。

「変換の作成」ウィザードが表示されます。ここでは、ターゲット・プロジェクトの選択、変換の名前指定、パッケージ名の指定変更 (HATS リッチ・クライアント・プロジェクトの場合)、説明の記述、変換定義の保存場所の参照を実行できます。これらの項目を指定したら、「**次へ**」をクリックします。

「**画面の選択**」パネルには、変換の土台となるホスト画面を選択するオプションがあります。「**画面の選択**」セクションのプルダウン・メニューから、「**ホスト端末画面を使用**」 または「**以前にキャプチャーした画面を使用**」を選択できます。ホスト画面の特定のフィールドを強調表示するには、「**強調表示するフィールド**」の隣にある各オプションを選択します。入力フィールドが画面上で定義されている位置を表示するには、「**入力**」チェック・ボックスを選択します。保護フィールドを表示する場合は、「**保護**」チェック・ボックスを選択します。隠しフィールドを強調表示する場合は、「**非表示**」チェック・ボックスを選択します。入力フィールド、保護フィールド、または隠しフィールドの強調表示色を変更する方法については、149 ページの『HATS プリファレンスの使用』を参照してください。選択が完了したら、「**次へ**」をクリックします。

© Copyright IBM Corp. 2003, 2015

選択できるレンダリング・パターンは、「**ブランク**」、「**デフォルト・レンダリング**」、「**デフォルト・レンダリングを使用して事前設定**」、「**フィールドを使用して事前設定**」の 4 つです。

ブランク

新規のブランク変換を作成します。コンポーネントを後で個別に追加する場合は、「デフォルト・レンダリングを挿入」ウィザードまたは「ホスト・コンポーネントを挿入」ウィザードを使用します。

デフォルト・レンダリング

実行時に、デフォルトのレンダリングで表示する新規の変換画像を作成します。デフォルト以外のレンダリング・セットを使用したり、レンダリングされる画面の領域を変更したり、「ホスト・コンポーネントを挿入」ウィザードを使用して追加コンポーネントを加えたりする場合は、このパターンを使用します。「**オプション**」をクリックし、「レンダリング・オプション」パネルで次の設定値を構成します。

デフォルト・レンダリング・セットを使用

実行時に変換画面をレンダリングするときに、アプリケーションのデフォルト・レンダリング・セットを使用する場合は、このボックスを選択します。特定のレンダリング・セットを選択して常に使用する場合は、このボックスをクリアします。

デフォルト・レンダリングを使用して事前設定

デフォルト・レンダリングを使用して生成されたコンポーネント・タグが事前に入力された新規の変換を作成します。この変換は、「ホスト・コンポーネントを編集」ウィザードを使用してカスタマイズできます。「**オプション**」をクリックし、「レンダリング・オプション」パネルで次の設定値を構成します。

デフォルト・レンダリング・セットを使用

実行時に変換画面をレンダリングするときに、アプリケーションのデフォルト・レンダリング・セットを使用する場合は、このボックスを選択します。特定のレンダリング・セットを選択して常に使用する場合は、このボックスをクリアします。

空の保護フィールドを含める

空の保護フィールド用にタグ (Web 変換の場合) またはコンポーネント・レンダリング・コンポジット (RCP 変換の場合) を生成するには、このボックスを選択します。ほとんどの場合、デフォルトの設定 (これらのフィールドのタグまたはコンポジットを生成しない) で十分であり、変換に関係する他の制御を移動する際の柔軟性を高めることができます。

注:

一般に、事前に入力された変換を作成する場合は、表示可能なできるだけ多くの情報を含む画面を取り込みます。例えば、テーブルを含む画面の変換を作成する場合は、データが入力された状態のテーブル全体を含む画面を取り込みます。これにより、テーブルのすべてのフィールド用にタグ (Web 変換の場合) およびコンポジット (RCP 変換の場合) が生成されます。

フィールドを使用して事前設定

ラベルと入力フィールドを使用して事前設定した変換を新規作成します。これは、「ホスト・コンポーネントを編集」ウィザードでカスタマイズできます。「**オプション**」をクリックし、「レンダリング・オプション」パネルで次の設定値を構成します。

空の保護フィールドを含める

空の保護フィールド用にタグ (Web 変換の場合) またはコンポーネント・レンダリング・コンポジット (RCP 変換の場合) を生成するには、このボックスを選択します。ほとんどの場合、デフォルトの設定 (これらのフィールドのタグまたはコンポジットを生成しない) で十分であり、変換に関係する他の制御を移動する際の柔軟性を高めることができます。

注:
一般に、事前に入力された変換を作成する場合は、表示可能なできるだけ多くの情報を含む画面を取り込みます。例えば、テーブルを含む画面の変換を作成する場合は、データが入力された状態のテーブル全体を含む画面を取り込みます。これにより、テーブルのすべてのフィールド用にタグ (Web 変換の場合) およびコンポジット (RCP 変換の場合) が生成されます。

選択したパターンを使用して変換画面を表示するには、「**プレビュー**」ボタンを使用します。

パターンのレイアウトが完了したら、変換画面の内部でコンポーネントを配置できます。 「画面カスタマイズ」ウィザードを起動する場合は、「**完了したら「画面カスタマイズを作成」ウィザードを開始**」チェック・ボックスを選択して、「**完了**」をクリックします。詳細については、 111 ページの『デフォルト・レンダリング』を参照してください。

変換の編集

HATS Web プロジェクト内の変換画面は JSP ファイルで、デフォルトでは Page Designerを使用して編集します。 HATS リッチ・クライアント・プロジェクト内の変換は SWT コンポジットで、Java ビジュアル・エディター・フィーチャーが Rational SDP とともにインストールされていると想定すると、デフォルトでは Visual Editor を使用して編集されます。

Web プロジェクトの変換の編集

作成した変換画面を参照するには、「**HATS Projects**」ビューで、「**Web コンテンツ/変換**」フォルダーを展開します。 Rational SDP に組み込まれている Page Designerを使用して変換画面を編集するには、変換画面の名前をダブルクリックします。使用可能なその他のエディターを参照するには、変換画面の名前を右クリックして、「**オープン先を指定**」を選択します。 メニュー・バーの「**ヘルプ**」>「**ヘルプ目次**」を選択して「Page Designer」を検索することにより、Page Designerに関する Rational SDP 資料を参照してください。

注: HATS では、Web 変換の編集に対しては Visual Editor をサポートしていません。

以下の制限事項は、変換の編集に適用されます。

- 変換は UTF-8 でエンコードされている必要があります。
- 名前が HATS、hats、または Hats で始まる JSP 変数、CSS クラス、HATSForm、またはその他のオブジェクトは使用しないでください。これらの名前は HATS 用に予約されています。
- 入力フィールドを <HATS:form> タグの外にドラッグすると、この入力フィールドは処理されず、サーバーに実行依頼されません。

注: ある HATS プロジェクトから別の HATS プロジェクトに変換をコピーした場合、その変換に関連付けられている画面キャプチャー・ファイルはコピーされません。コピーされた変換は、関連画面キャプチャー・ファイルがなくても実行時に機能します。ただし、関連画面キャプチャー・ファイルがないと、コピーされた変換を HATS Toolkit でプレビューすることはできません。

コピーした変換をプレビューするには、関連付けられている画面キャプチャー・ファイルを「**画面キャプチャー**」フォルダーで選択して、これをコピーする必要があります。画面キャプチャー・ファイルをコピーしたら、前の画面に戻り、「**変換**」フォルダーからコピーした変換を選択して、変換を開きます。「**デザイン**」タブに移動して、ウィンドウ内の任意の場所で右クリックします。「**HATS ツール**」>「**Host Component を編集**」を選択します。ウィザードが表示されたら、ドロップダウン・メニューから画面を選択して、「**完了**」をクリックします。

変換を編集する際、「**プロパティー**」ビューを使用すると、 HATS コンポーネントやその他のコントロール (ボタン、イメージ、リンクなど) のプロパティーを変更できます。「プロパティー」ビューは、変換エディターの下のほうにあります。「**プロパティー**」ビューは、HATS メニューから「**ウィンドウ (Window)**」>「**ビューの表示 (Show View)**」>「**プロパティー**」を選択することでもアクセスできます。

また、「**パレット**」ビューを使用して、HATS コンポーネントを変換に追加することもできます。ドロップダウン・メニューからコンポーネントを選択して、それを変換にドラッグできます。また、「**パレット (Palette)**」にはさまざまなタグ (HTML、フォーム・タグ、JSP、Struts など) が含まれているので、編集を容易に行うことができます。このビューは、メニュー・バーから「**ウィンドウ (Window)**」>「**ビューの表示 (Show View)**」>「**パレット (Palette)**」を選択することでも起動できます。

注: カスタム・コンポーネント、ENPTUI、およびライト・ペン・コンポーネントは、Page Designerの「**パレット**」ビューの「HATS コンポーネント」セクションには表示されません。これらのコンポーネントを使用可能にすると、Rational SDP のメニュー・バーで「**HATS ツール**」>「**ホスト・コンポーネントを挿入**」をクリックすることにより、これらのコンポーネントにアクセスできます。

次のセクションでは、Page Designer の各タブを説明します。

デザイン

Page Designerの「**デザイン**」タブには、変換を変更したときの変換の現行WYSIWYG ビューが表示されます。 また「**デザイン**」タブでは、ビジュアル・コンポーネント・タグ (VCT) に Cascading Style Sheet (CSS) が適用されます。

フリー・レイアウト・テーブルを使用すると、コンポーネントやウィジェットをPage Designer内で自由に配置できます。 このエディターでは、必要なテーブル・セルを作成して自動的にコンポーネントが位置合わせされます。フリー・レイアウト・テーブルを使用可能にするには、メニュー・バーで「**表示**」>「**フリー・レイアウト・テーブル**」を選択します。 コンポーネントを「パレット」から変換までドラッグすることもできます。挿入するコンポーネントは、テーブル・セル当たり 1 つだけにすることをお勧めします。HATS:DefaultRendering またはコンポーネント・タグで定義された領域の中央に、HATS コンポーネントまたは他のタイプのコンポーネントを追加することはできません。 フリー・レイアウト・テーブルは、HATS の空の新規変換のプリファレンスとして設定することが可能です。詳しくは、 149 ページの『HATS プリファレンスの使用』を参照してください。

「**デザイン**」タブに JSP ファイルを表示すると、ファイルの左上に JSP 記号が現れます。これらの記号を非表示にするには、「**ウィンドウ**」>「**設定**」の順に移動して、「**Web**」プリファレンスを探します。「**Web**」>「**ページ・デザイン**」>「**外観**」>「**編集記号**」の順に移動して、「**JSP**」チェック・ボックスのチェック・マークを外します。

注: GB18030 各国語文字を使用する場合の考慮事項については、 510 ページの『コード・ページ 1388 (GB18030) の使用』を参照してください。

一方、「**デザイン**」タブでは、HATS Toolkit メニュー・バーの「HATS ツール」メニューまたは「パレット」を使用して、変換に対して項目を挿入するか、変換の項目を編集できます。

「**HATS ツール**」メニューには以下の項目がありますが、これらの各項目からは変換ウィザードが起動します。各ウィザードの詳細については、 213 ページの『「変換」ウィザード』を参照してください。

- **ホスト・コンポーネントを挿入**
- **ホスト・コンポーネントを編集**
- **Dojo 編集用に変換** (このツールに関連する GUI ウィザードはありません)
- **デフォルト・レンダリングを挿入**
- **デフォルト・レンダリングを編集**
- **タブ付きフォルダーを挿入**
- **マクロ・キーを挿入**
- **グローバル変数を挿入**
- **オペレーター情報域を挿入**
- **ホスト・キーパッドを挿入**
  - **デフォルト・キーパッド**
  - **カスタム・キーパッド**
  - **個別キー**

- **アプリケーション・キーパッドの挿入**
  - **デフォルト・キーパッド**
  - **カスタム・キーパッド**
  - **個別キー**
- **すべてのホスト・コンポーネントを挿入**
- **格納済み画面を挿入**
- **統合オブジェクト・プロパティーの挿入**
  - **入力**
  - **出力**
- **HATS アプリケーションへの転送の挿入**

最後の 2 つのメニュー項目 (「**統合オブジェクト・プロパティーを挿入**」と「**HATS アプリケーションへの転送を挿入**」) については、 403 ページの『第 13 章 統合オブジェクトの使用』で説明します。

「パレット」の「HATS コンポーネント」ドロワーには、以下のコンポーネントが格納されています。

- **コマンド行**
- **フィールド**
- **ファンクション・キー**
- **入力フィールド**
- **ヒント付き入力フィールド**
- **項目の選択**
- **選択リスト**
- **サブファイル**
- **テーブル**
- **テーブル (フィールド)**
- **テーブル (可視)**
- **テキスト**
- **URL**

注: 変換を編集するときに、連続する 2 つのテキスト文字の間に HATS コンポーネントを挿入しようとすると、カーソル位置にタグが直接挿入されない場合があります。

イメージをプロジェクトに追加する場合は、イメージをプロジェクトの Web Content/Common/Images ディレクトリーにインポートすることをお勧めします。イメージをインポートするには、「**ファイル**」>「**インポート**」>「**一般**」>「**ファイル・システム**」をクリックして「インポート」ウィザードを開きます。「**ソース・ディレクトリー (From directory)**」フィールドで、インポートするイメージ・ソース・ファイルの場所を選択します。「**ターゲット・フォルダー (Into folder)**」として *project_name*/Web Content/Common/Images ディレクトリーを選択します。イメージ・ソース・ファイルをインポート後、「**イメージ**」フォルダーを右マウス・ボタンでクリックし、右下のウィンドウの「**サムネール (Thumbnail)**」タブで「**サムネ**

ールを表示」を選択します。イメージは、変換の「**デザイン**」タブ表示にドラッグ・アンド・ドロップしてコピーできます。

ソース

「**ソース**」タブには、ホスト画面からホスト・コンポーネントを抽出するために必要な *transformation_name*.jsp ファイル内の HTML タグと JSP タグ、これらのホスト・コンポーネントを提供するために選択したウィジェット、および変換に追加したその他のすべての項目が表示されます。Page Designerの別のタブで変更を行うと、ソース・ファイルのタグ内に表示されるタグと属性がそれに合わせて変更されます。

ソース・ファイル内のタグと属性を直接変更したり、HATS Toolkit メニュー・バーの「**HATS ツール**」ドロップダウン・メニューを使用して項目を挿入することもできます。「**ソース**」タブで挿入できる項目は、「**デザイン**」タブにリストされた項目と同じです。ソース・ファイル内で、メニュー項目の 1 つを挿入する位置にカーソルを置き、「**HATS ツール**」ドロップダウン・メニューからオプションを選択します。

ホスト・コンポーネントおよびそのレンダリング・ウィジェットが挿入されると、「**HATS ツール**」ドロップダウン・メニューの「**Host Component を編集**」 ツールバー・オプションを使用してホスト・コンポーネントおよびウィジェットを変更できます。「**Host Component を編集**」をクリックする前に、カーソルを必ず <HATS:Component> タグ内に置いてください。

「**ソース**」タブに表示されたファイルを変更すると、その変更内容は Page Designer のその他のタブに反映されます。

分割

「**分割 (Split)**」タブには 2 つのビューが表示されます。ツールバー・ボタンを使用して、「**設計**」ビューと「**ソース**」ビューの両方、または「**プレビュー**」ビューと「**ソース**」ビューの両方の表示を構成できます。

プレビュー

「**プレビュー**」タブでは、静的な HTML コンテンツを表示したり <HATS:Component > (およびすべての <HATS:*xxxx*>) タグをレンダリングする変換のブラウザー・プレビューが提供されます。このタブは「**デザイン**」タブに似ていますが、コンテンツの変更はできません。

リッチ・クライアント・プロジェクトの変換の編集

作成した変換画面を参照するには、「**HATS Projects**」ビューで、「**リッチ・クライアント・コンテンツ/変換**」フォルダーを展開します。

デフォルト・エディターを使用して変換を編集するには、変換の名前をダブルクリックします。使用可能なエディターを参照するには、変換画面の名前を右クリックして、「**アプリケーションから開く**」を選択します。

Java ビジュアル・エディター フィーチャー (別名 Visual Editor) が Rational SDP とともにインストールされている場合は、これを使用して変換を編集できます。このセクションの情報に加えて、「*HATS リッチ・クライアント・プラットフォー*

ム・プログラマーズ・ガイド」 の章『変換』を参照してください。また、ご使用の Rational SDP 製品の Knowledge Center も参照してください。例えば、http://publib.boulder.ibm.com/infocenter/radhelp/v8r5/index.jsp を参照し、`visual editor` を検索します。

Visual Editor の 2 つの領域は、通常は上段の領域として表示される「デザイン」ビューと、通常は下段の領域として表示される「ソース」ビューです。コンポーネントやウィジェットなどの項目は、「**パレット**」ビューで項目を選択して「デザイン」ビューにドロップすることにより、変換に追加されます。

「デザイン」ビューで、変換に少なくとも 1 つ以上のコンポーネント・レンダリングまたはデフォルト・レンダリングが含まれる場合、デフォルト・テンプレートのスタイルが適用されます。プロジェクトのデフォルト・テンプレートのみが適用されます。HATS では、変換と関連付けられた適用アクションに関連するテンプレートのスタイルを適用しようとはしません。変換画面の編集中にデフォルト・テンプレートを変更する場合は、エディターを閉じて再オープンするまでスタイルの変更内容が表示されない可能性があります。

「パレット」ビュー

「**パレット**」ビューの HATS ドロワーには、ホスト・コンポーネントやその他のウィジェットのコンポジットを HATS SWT 変換に追加するときに使用できる以下の項目が収容されています。参照される各ウィザードについては、213 ページの『「変換」ウィザード』を参照してください。

コンポーネント
「ホスト・コンポーネントを挿入」ウィザードを起動して、特定のコンポーネントのレンダリングを挿入する場合は、この項目を使用します。

デフォルト・レンダリング
「デフォルト・レンダリングを挿入」ウィザードを起動して、デフォルト・レンダリング・コンポジットを挿入する場合は、この項目を使用します。

マクロ・キー
「マクロ・キーを挿入」ウィザードを起動して、クリックしたときにマクロを実行するボタンまたはリンクを挿入する場合は、この項目を使用します。

グローバル変数
「グローバル変数を挿入」ウィザードを起動して、グローバル変数の値を表示するか値の入力を求める場合は、この項目を使用します。

ホスト・キー
「ホスト・キーを挿入」ウィザードを起動して、クリックしたときにホスト・キーを送信するボタンまたはリンクを挿入する場合は、この項目を使用します。

アプリケーション・キー
「アプリケーション・キーを挿入」ウィザードを起動して、アプリケーション・キー機能に対応するボタンまたはリンクを挿入する場合は、この項目を使用します。

「プロパティー」ビュー

変換に追加されたコンポジットに変更を加えるには、「**プロパティー**」ビューを使用します。以下に示すタイプの変更を行うためのダイアログが用意されています。

- 関連する画面キャプチャーの変更 - デフォルト・レンダリングまたは個別のコンポーネント・レンダリングの場合に有効です。
- 画面領域の変更 - デフォルト・レンダリングまたは個別のコンポーネント・レンダリングの場合に有効です。
- 別のコンポーネントまたはウィジェットの選択 - 個別のコンポーネント・レンダリングの場合に有効です。
- コンポーネントまたはウィジェットの設定変更 - 個別のコンポーネント・レンダリングの場合に有効です。
- 別のデフォルト・レンダリング・セットの選択 - デフォルト・レンダリングの場合に有効です。

ヒント

- コンポーネントを HATS SWT 変換に追加するときに、追加するコンポーネントのサイズが大きすぎて収容できない場合は、変換コンポジットのサイズを変更することが必要な場合があります。 このためには、「デザイン」ビューで (追加したコンポーネントのみでなく) 変換コンポジットを選択し、「**プロパティー**」ビューで **size** 属性を更新します。変換コンポジットを、その内容に基づいて自動的にサイズ変更するには、該当する変換コンポジットを選択して右クリックし、「**レイアウトのカスタマイズ**」を選択します。さらに、「**コンポーネント**」タブで、「**設定済みサイズの復元**」ボタンをクリックします。
- タブ付きフォルダーのコンポジットを HATS SWT 変換に挿入するには、「**パレット**」ビューで「SWT コンテナー」フォルダーに置かれている SWT TabFolder ウィジェットを使用します。
- Visual Editor を使用して、フォント、画像、色などのリソースを HATS 変換またはテンプレートに手動で追加する場合は、これらのリソースをクリーンアップするためのコードも手動で追加する必要があります。詳しくは、103 ページの『HATS リッチ・クライアントの考慮事項と制限』を参照してください。
- ウィザードを起動してホスト・コンポーネントやウィジェットのコンポジットを HATS SWT 変換に追加した後で、「キャンセル」ボタン、「ウィンドウをクローズ」ボタン、または Esc キーを使用してこのウィザードを取り消すと、以下の例と同様に、ワークスペース・ディレクトリーのルートにある .metadata/.log ファイル内にトレースが書き込まれます。

```
!ENTRY org.eclipse.ve.java.core 2 0 2008-09-21 15:57:59.275
!MESSAGE Exception thrown.
!STACK 0
com.ibm.hats.jve.CancelOperationException
```

- ある HATS リッチ・クライアント・プロジェクトから別のプロジェクトに変換をコピーした場合、その変換に関連付けられている画面キャプチャー・ファイルはコピーされません。コピーした変換を表示しようとするか、コンポーネントを挿入しようとしても、表示する画面キャプチャーはありません。

  コピーした変換が機能するようにするには、関連付けられている画面キャプチャー・ファイルを「**画面キャプチャー**」フォルダーにコピーする必要があります。

画面キャプチャー・ファイルをコピーしたら、前の画面に戻り、「**変換**」フォルダーからコピーした変換を選択して、変換を開きます。

「**プロパティー**」**ビュー**で、変換に追加されたコンポーネントを選択して、画面キャプチャーの隣のボタンをクリックし、画面キャプチャーを選択します。

V7.0.0.2 より前のリリースの HATS で作成されたプロジェクトから変換と画面キャプチャーをコピーした場合、変換のソースでは、screenCapture Java クラス・メンバーが画面キャプチャー・ファイルの名前を参照していることを確認してください。各コンポーネント・レンダリング・コンポジットの setScreenCapture メソッドがこのフィールドを参照するため、各 setScreenCapture メソッド呼び出しを更新して、変換で screenCapture フィールドの正しい完全修飾 Java クラス名を使用する必要があります。例えば、RCPProjectA から RCPProjectB に変換をコピーする場合、新規の値 (変換でデフォルトの Java パッケージを使用するように設定していると想定) は RCPProjectB.transformations.<transformation name>.screenCapture です。

Web 変換とリッチ・クライアント変換の編集の違いの要約

以下のリストでは、Page Designer を使用した Web 変換の編集と、Visual Editor を使用したリッチ・クライアント変換の編集の主な違いについて要約しています。

- Web 変換を編集する場合、デフォルトでは Page Designerが使用されます。Java ビジュアル・エディター・フィーチャーが Rational SDP とともにインストールされていると想定すると、リッチ・クライアント変換の編集ではデフォルトで Visual Editor が使用されます。
- コンポーネントおよびウィジェットの Web 変換への挿入は、メニュー・バーの「**HATS ツール**」メニューまたは「**パレット**」ビューで実行します。 コンポーネントおよびウィジェットのリッチ・クライアント変換への挿入は、必ず「**パレット**」ビューで実行します。
- Web 変換の場合は、該当するウィザードの単一インスタンスの内部から複数のマクロ・キー、ホスト・キー、またはアプリケーション・キーを挿入できます。リッチ・クライアント変換の場合、複数のマクロ・キー、ホスト・キー、またはアプリケーション・キーを挿入するには、該当するウィザードを複数回起動して、毎回キーを 1 つずつ挿入します。
- タブ付きフォルダーを Web 変換に挿入するには、HATS の「タブ付きフォルダーを挿入」ウィザードを使用します。 リッチ・クライアント変換の場合は、SWT TabFolder ウィジェットを使用します。
- すべてのホスト・コンポーネントを Web 変換に挿入するには、「すべてのホスト・コンポーネントを挿入」ウィザードを使用するか、「変換の作成」ウィザードで「**デフォルト・レンダリングを使用して事前設定**」または「**フィールドを使用して事前設定**」のいずれかのパターンを使用します。リッチ・クライアント変換の場合は、必ず「変換の作成」ウィザードでパターンを使用してください。
- Web 変換上でのコンポーネントおよびウィジェットの変更は、メニュー・バーの「**HATS ツール**」メニューまたは「**プロパティー**」ビューで実行します。 リッチ・クライアント変換上でのコンポーネントおよびウィジェットの変更は、必ず「**プロパティー**」ビューで実行します。

「変換」ウィザード

ホスト・コンポーネントを挿入

「ホスト・コンポーネントを挿入」ウィザードを使用して、ホスト・コンポーネントを抽出する画面を選択します。また、テキストの周囲に長方形を描いて、ホスト・コンポーネントを抽出する画面領域を選択します。カーソルを画面上の一点に置き、左マウス・ボタンをクリックして押しながら別の点に移動して長方形を描きます。ウィザードの最下部のフィールドに、長方形の開始点および終了点の行番号とカラム番号が表示されます。行番号とカラム番号は、このフィールドにユーザーが入力することもできます。

ホスト画面の特定のフィールドを強調表示するには、「**強調表示するフィールド**」の隣にある各オプションを選択します。入力フィールドが画面上で定義されている位置を表示するには、「**入力**」チェック・ボックスを選択します。保護フィールドを表示する場合は、「**保護**」チェック・ボックスを選択します。隠しフィールドを強調表示したい場合は、「**非表示**」チェック・ボックスを選択します。入力フィールド、保護フィールド、または隠しフィールドの強調表示色を変更する方法については、149 ページの『HATS プリファレンスの使用』を参照してください。画面の開始点および終了点の行番号とカラム番号を選択したら、「**次へ**」をクリックして、選択した領域にあるホスト・コンポーネントのレンダリング・オプションを表示します。

HATS は、ホスト・コンポーネント・ウィジェットを備えています。提供されたホスト・コンポーネントおよびウィジェットのいずれかを選択することも、ユーザー固有のカスタムホスト・コンポーネントおよびウィジェットを作成することもできます。ホスト・コンポーネントおよびウィジェットの詳細については、223 ページの『第 9 章 コンポーネントおよびウィジェットの説明と設定』を参照してください。

「**コンポーネント**」リスト内で、いずれかのコンポーネントをクリックします。画面領域にコンポーネントが存在する場合は、「コンポーネント・プレビュー」ウィンドウにそのコンポーネントが表示されます。コンポーネントに対してテキスト置換を使用する場合は、コンポーネント・リストの右側にある「**テキスト置換**」アイコンをクリックし、設定を開いて編集します。テキスト置換の詳細については、195 ページの『テキスト置換』を参照してください。

対応する「**ウィジェット**」リスト内のウィジェットから、ホスト・コンポーネントのレンダリングに使用するウィジェットを選択します。ウィジェットを選択すると、「ウィジェット・プレビュー」ウィンドウに、最終 Web ページに表示されるウィジェットのプレビューが表示されます。「**大きいウィンドウでウィジェットをプレビュー**」アイコン (虫眼鏡) をクリックすると、ウィジェット・プレビューを拡大できます。

コンポーネントとウィジェットの個々のインスタンスのデフォルトのプロジェクト・レベルの設定は、「**コンポーネント**」リストと「**ウィジェット**」リストの右側にある「**コンポーネント設定**」アイコンと「**ウィジェット設定**」アイコンをクリックすることでオーバーライドできます。コンポーネントとウィジェットのデフォル

トのプロジェクト・レベルの設定は、プロジェクト・エディターの「**レンダリング**」タブを使用して構成されます。詳しくは、 111 ページの『レンダリング』を参照してください。

拡張レンダリング・オプションを指定する場合は、「**ウィジェット**」リストの右側にある「**レンダリングの拡張設定**」アイコンをクリックします。 プロジェクト・レベルのデフォルトの拡張レンダリング設定は、プロジェクト・エディターの「**レンダリング**」タブを使用して構成されます。詳しくは、 111 ページの『レンダリング』を参照してください。

HATS Web プロジェクトの場合、「**全ページのプレビュー**」をクリックすると、ページ内のすべてのコンポーネントおよび関連するテンプレートが表示されます。このプレビューでは、ページがユーザーに表示されるとおりに表示されます。

注: HATS コンポーネントを変換のフリー・レイアウト・テーブルに挿入すると、「**ホスト・コンポーネントを挿入**」ウィザードの「**全ページのプレビュー**」機能で、フリー・レイアウト・テーブルの上部に常に新規コンポーネントが表示されます。プレビューで誤った表示になる場合もありますが、ウィザードで「**完了**」をクリックすると、コンポーネントはテーブル内の正しい位置に挿入されます。

HATS リッチ・クライアント・プロジェクトの場合、「**ホスト・コンポーネントを挿入**」ウィザードでは、全ページのプレビューはサポートされていません。全ページのプレビューを参照するには、HATS Toolkit の以下のいずれかの領域に配置されているプレビュー機能を使用してください。

- 画面キャプチャー・エディターの「プレビュー」タブ
- ホスト端末の「プレビュー」タブ
- 「画面カスタマイズ」ウィザード/「アクションの選択」パネルの「プレビュー」ボタン
- 「新規変換」ウィザード/「レンダリング・オプション」パネルの「プレビュー」ボタン

「ホスト・コンポーネントを挿入」ウィザードを使用してカスタマイズできる設定については、 223 ページの『コンポーネントおよびウィジェットの設定』を参照してください。使用可能なウィジェットは、選択したホスト・コンポーネントにより異なります。 364 ページの表 2 に、既存の HATS ホスト・コンポーネントおよび対応するウィジェットを示しています。

HATS では、画面領域内にコンポーネントが検出されないと、「コンポーネントのプレビュー」ウィンドウに「指定した領域で *component_name* コンポーネントは見つかりませんでした。」というメッセージが表示されます。ここで、*component_name* は、「**コンポーネント**」リストで選択したコンポーネントを表します。 このメッセージが表示された場合は、コンポーネント全体を含む領域が選択されていない可能性があります。場合によっては、「**コンポーネント設定**」ダイアログを使用して、コンポーネントの設定をホスト・アプリケーションのコンポーネント表示方法に合わせて変更する必要があります。例えば、領域内にコマンド行コンポーネントがあり、ご使用のコマンド行ではトークン ==> の代わりに >>> を使用

している場合があります。この場合は、コマンド行コンポーネントのトークン属性を変更して、正しいトークンでコマンド行コンポーネントを探すように修正できます。

コンポーネントとウィジェットを選択したら、「**完了**」を選択します。

ホスト・コンポーネントを編集Web のみ

「ホスト・コンポーネントを編集」ウィザードを使用すると、「**画面の選択**」ドロップダウン・リストで別の画面を選択して、その画面からコンポーネントを検索できます。また、テキストの周囲に長方形を描くことで、画面上で別の領域を選択して、この中からホスト・コンポーネントを抽出できます。

「**ホスト・コンポーネントを編集**」には、「**ホスト・コンポーネントを挿入**」と同じ機能があります。これを使用すると、既に挿入済みのタグ (例えば、既に配置済みの <HATS:Component> タグなど) を編集できます。詳しくは、 213 ページの『ホスト・コンポーネントを挿入』を参照してください。

注: リッチ・クライアント変換の場合は、「**プロパティー**」ビューを使用してホスト・コンポーネント・コンポジットを編集してください。

Dojo 編集用に変換 Web のみ

ホスト・コンポーネントのレンダリング用に以前に追加しておいた HATS Dojo ウィジェットをカスタマイズするには、この HATS ツールを使用します。このツールは、HATS Dojo ウィジェットによって使用されるタグを変更することで、ユーザー独自のカスタマイズを可能にします。HATS Dojo ウィジェットのカスタマイズ方法の例は、*「HATS Web アプリケーション・プログラマーズ・ガイド」* の『HATS Dojo ウィジェットのカスタマイズ』を参照してください。

デフォルト・レンダリングを挿入

「デフォルト・レンダリングを挿入」ウィザードを使用すると、使用する画面を「**画面**」ドロップダウン・リストから選択して、レンダリング・セットを「**レンダリング・セットの選択**」ドロップダウン・リストから選択できます。デフォルト・レンダリングを挿入すると、定義された領域では、項目がこのレンダリング・セットの順序でレンダリングされます。詳しくは、 111 ページの『デフォルト・レンダリング』を参照してください。

ウィザードの最下部のフィールドに、長方形の開始点および終了点の行番号とカラム番号が表示されます。行番号とカラム番号は、このフィールドにユーザーが入力することもできます。ホスト画面の特定のフィールドを強調表示するには、「**強調表示するフィールド**」の隣にある各オプションを選択します。入力フィールドが画面上で定義されている位置を表示するには、「**入力**」チェック・ボックスを選択します。保護フィールドを表示する場合は、「**保護**」チェック・ボックスを選択します。隠しフィールドを強調表示する場合は、「**非表示**」チェック・ボックスを選択します。入力フィールド、保護フィールド、または隠しフィールドの強調表示色を変更する方法については、149 ページの『HATS プリファレンスの使用』を参照してください。画面の開始点と終了点の行番号とカラム番号を選択後、「**完了**」をクリックします。これにより、このセクションが「**設計 (Design)**」タブ上の画面に挿入されます。

デフォルト・レンダリングを編集Web のみ

デフォルト・レンダリングを編集するには、「**設計 (Design)**」タブ上で編集するセクションをクリックし、「**HATS ツール**」メニューから「**デフォルト・レンダリングを編集**」を選択します。これにより、「デフォルト・レンダリングを編集」ウィザードが起動します。ここでは、「**画面**」ドロップダウン・リストから編集対象の画面を選択して、「**レンダリング・セットの選択**」ドロップダウン・リストからレンダリング・セットを選択できます。デフォルト・レンダリングを編集すると、定義された領域では、項目がこのレンダリング・セットの順序でレンダリングされます。詳しくは、 111 ページの『デフォルト・レンダリング』を参照してください。

ウィザードの最下部のフィールドに、長方形の開始点および終了点の行番号とカラム番号が表示されます。行番号とカラム番号は、このフィールドにユーザーが入力することもできます。ホスト画面の特定のフィールドを強調表示するには、「**強調表示するフィールド**」の隣にある各オプションを選択します。入力フィールドが画面上で定義されている位置を表示するには、「**入力**」チェック・ボックスを選択します。保護フィールドを表示する場合は、「**保護**」チェック・ボックスを選択します。隠しフィールドを強調表示する場合は、「**非表示**」チェック・ボックスを選択します。入力フィールド、保護フィールド、または隠しフィールドの強調表示色を変更する方法については、 149 ページの『HATS プリファレンスの使用』を参照してください。画面の開始点と終了点の行番号とカラム番号を選択後、「**完了**」をクリックします。これにより、このセクションが「**設計 (Design)**」タブ上の画面に挿入されます。

注: リッチ・クライアント変換の場合は、「**プロパティー**」ビューを使用してデフォルト・レンダリング・コンポジットを編集してください。

タブ付きフォルダーを挿入Web のみ

注: HATS タブ付きフォルダー・サポートは、HATS V9.5 では推奨されません。現時点では、タブ付きフォルダーは引き続きサポートされていますが、IBM では、今後の製品リリースでこの機能を除去する場合があります。代替方法として以下の方法があります。

- Dojo TabContainer ウィジェットでタブを作成します。詳しくは、*「HATS Web アプリケーション・プログラマーズ・ガイド」* の『Dojo TabContainer ウィジェットの使用』を参照してください。
- カスタムの DIV セクションと JavaScript を作成して、個々の DIV セクションの可視性を制御します。ツールやサンプルは、インターネットのいたるところで入手できます。

「**タブ付きフォルダーを挿入**」は、Web ページにタブ付フォルダーを挿入する場合に使用します。タブ付きフォルダーは、Web ページに表示するウィジェットやコンポーネントを整理する際に役立ちます。「**タブ付きフォルダーを挿入**」では、フォルダーに必要なタブの数を指定します。タブ付きフォルダーと格納済み画面機能を組み合わせると、タブごとに異なるホスト画面を持つフォルダーを持つことができます。フォルダー内の各タブについて、次の設定も指定します。

- タブのラベル・テキスト
- タブに表示するホスト・コンポーネント

- タブを選択しないときのタブの背景色
- タブを選択したときのタブの背景色

「**タブ拡張オプション**」で「**デフォルト値を使用**」チェック・ボックスをクリアすると、次の設定を指定できます。

- タブを選択したときのタブ・ページ上でのテキストの色
- タブを選択していないときのタブ・ページ上でのテキストの色
- タブにカーソルを置いたときのタブ・ページの色

「**フォルダー拡張オプション**」で「**デフォルト値を使用**」チェック・ボックスをクリアすると、次の設定を指定できます。

- タブの高さ (ピクセル)
- フォルダーの幅 (ピクセル)
- フォルダーの高さ (ピクセル)
- フォルダー・アウトラインの色

「プレビュー」ウィンドウに、選択した項目により設定されたタブ付きフォルダーのプレビューが表示されます。このプレビューには、タブ付きフォルダーに挿入されたコンポーネントは表示されません。フォルダーに対して行った変更 (サイズや色) のみが表示されます。

「**全ページのプレビュー**」をクリックすると、ページ内のすべてのコンポーネントおよび関連するテンプレートが表示されます。このプレビューでは、ページがユーザーに表示されるとおりに表示されます。

フォルダー内の各タブに対してタブ付きフォルダー・オプションをすべて定義したら、「**OK**」をクリックします。

タブ付きフォルダーを作成した後は、「タブ付きフォルダーを挿入」ウィザードに戻って変更を行うことはできません。出力は HTML (および HATS コンポーネント・タグ) 形式なので、ソースを編集する必要があります。

注:

1. タブ付きフォルダーは複数の HTML タグで構成されるため、セクションを個別に移動させることはお勧めしません。タブ付きフォルダーの場所を変更したい場合は、タブ付きフォルダーを含むセルを選択し、そのセルを希望の場所に移動します。
2. リッチ・クライアント変換の場合、タブ付きフォルダーのコンポジットを挿入するには、「**パレット**」ビューで「SWT コンテナー」フォルダーに置かれている SWT TabFolder ウィジェットを使用します。

マクロ・キーを挿入

「マクロ・キーを挿入」を使用すると、ユーザーは、ボタンやリンクをクリックしたり、ドロップダウン・リストからマクロを選択することにより、変換上でマクロを実行できます。例えば、変換により、ログオン・マクロのボタンがあるログオン画面を表示できます。そのボタンをクリックすると、マクロが実行されてユーザー ID とパスワードが入力され、ユーザーの確認が必要な次の画面に移動します。

マクロを変換に追加するには、追加するマクロを選択します。 Web プロジェクトの場合は、ウィザードの同一インスタンス内で複数のマクロを選択できます。リッチ・クライアント・プロジェクトの場合、選択できるマクロはウィザードのインスタンスごとに 1 つのみです。変換のマクロの表示方法および開始方法も定義する必要があります。次のオプションのいずれかを選択します。

- ボタン
- リンク
- ドロップダウン・リスト Web のみ

注: HATS では、マクロの説明をレンダリング・オプションの変換に挿入するテキストとして使用します。ただし、「**ソース**」ビューでテキストを編集して、名前を変更することができます。

グローバル変数を挿入

「グローバル変数を挿入」を使用すると、次のオプションのいずれかを行うことができます。

- グローバル変数をテキストとして変換に挿入する。
- グローバル変数に関するプロンプトを入力フィールドとともに表示する。

値を表示する定義済みグローバル変数を、「**名前**」ドロップダウン・リストから選択します。

次に、グローバル変数を実行時にどのように表示するかを選択します。「**グローバル変数値を静的なテキストとして表示**」を選択した場合は、「**拡張**」ボタンと「**変数は索引付き**」チェック・ボックスをクリックできます。次に、ラジオ・ボタンを使用して、挿入する索引をすべてにするか 1 つにするかを指定します。「**索引をすべて表示**」ラジオ・ボタンをクリックした場合は、ドロップダウン・リストから「**分離文字**」を選択します。また、「**単一の索引を表示**」ラジオ・ボタンを選択した場合は、挿入する索引の数を指定します。「**共有**」チェック・ボックスを選択すると、共用リストからグローバル変数が取得されます。

入力ボックスを使用しているユーザーにプロンプトを表示したい場合は、「**グローバル変数に関するプロンプトを入力ボックスとともに表示する。(Prompt for global variable with input box.)**」を選択します。入力フィールドに初期値を挿入するには、「**グローバル変数の初期値を設定**」を選択します。「**パスワード・フィールドとしてマスク**」を選択すると、ユーザーが入力フィールドに挿入する内容がマスクされます。

オペレーター情報域を挿入Web のみ

「**オペレーター情報域**」は、「**プロジェクト設定**」の「**レンダリング**」タブに指定した使用可能な情報を表示する場合に選択します。詳しくは、122 ページの『オペレーター情報域』を参照してください。

ホスト・キーパッドを挿入

ホスト・キーパッドは、さまざまな方法で挿入できます。デフォルト・ホスト・キーパッドまたはカスタム・ホスト・キーパッドを挿入することもできますし、変換のキーを個々に選択することもできます。

デフォルト・キーパッドWeb のみ

デフォルト・ホスト・キーパッドを変換に追加するには、「**ホスト・キーパッドを挿入**」>「**デフォルト・キーパッド (Default Keypad)**」を選択します。挿入されたホスト・キーパッドは、HATS アプリケーションにデフォルト・ホスト・キーパッドが表示されるようにプロジェクト設定で構成されている場合にのみ、「**プレビュー**」および「**設計 (Design)**」表示に表示されます。プロジェクトを構成してデフォルトのホスト・キーパッドを表示し、どのキーを対象にするかを定義するには、「**HATS Projects**」ビューに移動して、HATS プロジェクトの「**プロジェクト設定**」をダブルクリックし、「**レンダリング**」タブを選択して、「**ホスト・キーパッド**」をクリックします。

カスタム・キーパッドWeb のみ

カスタム・ホスト・キーパッドを変換に追加するには、「**ホスト・キーパッドを挿入**」>「**カスタム・キーパッド (Custom Keypad)**」を選択します。カスタム・ホスト・キーパッドでは、デフォルト・ホスト・キーパッド設定で指定したものと同じキーが使用されます。これは、ご使用の HATS プロジェクトの「**プロジェクト設定**」に定義されています。カスタム・ホスト・キーパッドを使用すると、「**設計 (Design)**」表示でボタンやリンクを強調表示し、右マウス・ボタンでクリックして「**プロパティー**」を選択することにより、個々のキーパッド・ボタンやリンクの属性を編集できます。

個々のキー

このオプションを使用して、個別のホスト・キーを変換に追加できます。「**ホスト・キーを挿入**」パネルで、挿入するホスト・キーを選択し、これをボタンまたはリンクとして表示するかどうかを指定して、「**OK**」をクリックします。Web プロジェクトの場合は、ウィザードの同一インスタンス内で複数のホスト・キーを選択できます。リッチ・クライアント・プロジェクトの場合、選択できるホスト・キーはウィザードのインスタンスごとに 1 つのみです。挿入後にボタンまたはリンクを編集するには、「**プロパティー**」タブを使用します。

アプリケーション・キーパッドの挿入

アプリケーション・キーパッドは、さまざまな方法で挿入できます。デフォルト・アプリケーション・キーパッドまたはカスタム・アプリケーション・キーパッドを挿入することもできますし、変換のキーを個々に選択することもできます。

デフォルト・キーパッドWeb のみ

デフォルト・アプリケーション・キーパッドをテンプレートに追加するには、「**アプリケーション・キーパッドを挿入**」>「**デフォルト・キーパッド (Default Keypad)**」を選択します。デフォルトのアプリケーション・キーパッドを構成して、どのキーを表示するかを定義するには、「**HATS Projects**」ビューに移動して、HATS プロジェクトの「**プロジェクト設定**」をダブルクリックし、「**レンダリング**」タブを選択して、「**アプリケーション・キーパッド**」をクリックします。

カスタム・キーパッドWeb のみ

カスタム・アプリケーション・キーパッドをテンプレートに追加するには、「**アプリケーション・キーパッドを挿入**」>「**カスタム・キーパッド (Custom Keypad)**」を選択します。カスタム・アプリケーション・キーパッドでは、デフォルト・アプ

リケーション・キーパッド設定で指定したものと同じキーが使用されます。これは、ご使用の HATS プロジェクトの「**プロジェクト設定**」に定義されています。カスタム・アプリケーション・キーパッドを使用すると、「**設計 (Design)**」表示でボタンやリンクを強調表示し、右マウス・ボタンでクリックして「**プロパティー**」を選択することにより、個々のキーパッド・ボタンやリンクの属性を編集できます。

個別キー

このオプションを使用して、個別のアプリケーション・キーを変換に追加できます。「**アプリケーション・キーを挿入**」パネルで、挿入するアプリケーション・キーを選択し、これをボタンまたはリンクとして表示するかどうかを指定して、「**OK**」をクリックします。Web プロジェクトの場合は、ウィザードの同一インスタンス内で複数のアプリケーション・キーを選択できます。リッチ・クライアント・プロジェクトの場合、選択できるアプリケーション・キーはウィザードのインスタンスごとに 1 つのみです。挿入後にボタンまたはリンクを編集するには、「**プロパティー**」タブを使用します。

すべてのホスト・コンポーネントを挿入Web のみ

特定の画面キャプチャーまたはカスタマイズした画面から、指定可能なすべてのホスト・コンポーネントを変換に追加するには、「**すべてのホスト・コンポーネントを挿入**」を選択します。「**画面の選択**」ドロップダウン・メニューから該当するプロジェクト画面を選択します。ホスト画面の特定のフィールドを強調表示するには、「**強調表示するフィールド**」の隣にある各オプションを選択します。入力フィールドが画面上で定義されている位置を表示するには、「**入力**」チェック・ボックスを選択します。保護フィールドを表示する場合は、「**保護**」チェック・ボックスを選択します。隠しフィールドを強調表示したい場合は、「**非表示**」チェック・ボックスを選択します。入力フィールド、保護フィールド、または隠しフィールドの強調表示色を変更する方法については、149 ページの『HATS プリファレンスの使用』を参照してください。選択が完了したら、「**OK**」をクリックします。

追加したホスト・コンポーネントでは、HATS「**プロジェクト設定**」の「**レンダリング**」タブの「**レンダリング・セット (Rendering Sets)**」で指定したデフォルト・レンダリング・セットが使用されます。

格納済み画面を挿入Web のみ

格納済み画面を変換に追加するには、「**格納済み画面を挿入**」を選択します。格納済み画面名を選択し、「**全フィールド (無保護および保護)**」または「**共用**」を選択します。例えば、1 つの変換内で複数の格納済み画面を結合して、すべての画面のフィールドを持つ 1 つの画面を作成することができます。

複数の画面の結合方法については、415 ページの『第 14 章 画面の結合』を参照してください。

変換のプレビュー

前述したように、「ホスト・コンポーネントを挿入」ウィザードまたは「タブ付きフォルダーを挿入」ウィザードでは、ウィジェットを変換画面上での表示状態でプレビューできます。

変換をプレビューする別の方法として、画面キャプチャーを使用する方法があります。プロジェクト内で作成した変換ごとに、関連の画面キャプチャーがあります。画面キャプチャーを参照するには、「**HATS Projects**」ビューの「**画面キャプチャー**」フォルダーを展開します。 取り込み済み画面名をダブルクリックすると、「**ホスト画面**」タブおよび「**プレビュー**」タブの両方で、取り込んだ画面のビューを表示できます。「**ホスト画面**」タブには、ホスト上に表示される画面が表示されます。「**プレビュー**」タブには、変換のレンダリング方法、および変換に関連するテンプレートが表示されます。

プレビューの生成に使用される変換とテンプレートは、プロジェクトで定義された画面カスタマイズに基づいています。「**プレビュー**」では、使用可能な画面カスタマイズのリストをスキャンします。画面キャプチャーに一致する画面カスタマイズが検出されると、変換 (の他に関連のテンプレート) が適用される最初のアクションを使用してプレビューがレンダリングされます。一致する画面カスタマイズがみつからなかった場合は、デフォルトのテンプレートおよび変換がプレビューに適用されます。

注:

1. 「**プレビュー**」タブでは、画面認識イベントで定義された「**次の画面**」は処理されません。詳しくは、 197 ページの『次の画面』を参照してください。
2. 「**プレビュー**」タブでは、ポートレット・プロジェクトの変換画面が正しく表示されません。制限事項としては、イメージや CSS スタイルが正しく表示されないことが挙げられます。変換の内容によっては、JavaScript エラーも発生する場合があります。

ホスト・キーパッド

ホスト・キーパッドには、F1、F2、Clear など、ホスト・キーを表すボタンがあり、これらはホスト画面上での機能を制御するときに使用します。

HATS では、デフォルトではホスト・キーパッドは表示されません。ホスト・キーパッドを表示するには、「**HATS Projects**」ビューに移動して HATS プロジェクトの「**プロジェクト設定**」をダブルクリックし、「**レンダリング**」タブを選択して「**ホスト・キーパッド**」をクリックします。 これにより、キーパッドを表示するチェック・ボックスを選択したり、表示するキーを選択できるようになります。

注:

1. カスタム・キーパッドや個々のキーは、個々のタグまたはコンポジットを使用して定義するため、変換のエディターには表示されます。
2. ホスト画面上にあるファンクション・キーを説明するテキストは、デフォルト変換を使用してボタンまたはリンクに変換されます。これらは、変換されたホスト画面区域にホスト・キーパッドとは別に表示されます。

キーボード・サポートおよびキーパッドの設定の詳細については、 107 ページの『第 5 章 HATS プロジェクトの変更』を参照してください。

第 9 章 コンポーネントおよびウィジェットの説明と設定

HATS が提供するホスト・コンポーネントとウィジェットは、ホスト画面のエレメントを、Web ページで表示できるオブジェクトに変換するために使用されます。一部のコンポーネント設定およびウィジェット設定は、HATS Toolkit のウィザードやエディターを使用して変更できます。この章では、HATS ホスト・コンポーネント、ウィジェット、および変更可能な設定について説明します。

HATS ウィザードを使用して、ユーザー独自のカスタム・コンポーネントおよびウィジェットを作成することもできます。ウィザードを起動するには、「HATS」メニューから「**HATS**」>「**新規**」>「**コンポーネント**」または「**HATS**」>「**新規**」>「**ウィジェット**」を選択します。「**ファイル**」>「**新規**」>「**HATS コンポーネント**」または「**ファイル**」>「**新規**」>「**HATS ウィジェット**」を選択して、ウィザードを起動することもできます。

ウィザードを使用してカスタム・コンポーネントおよびウィジェットを作成する方法の詳細については、ご使用のアプリケーション環境に応じて、「*HATS Web アプリケーション・プログラマーズ・ガイド*」または「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

コンポーネントおよびウィジェットの設定

HATS が提供するコンポーネントおよびウィジェットには、変更可能なデフォルト設定があり、プロジェクト・エディターを使用するプロジェクト全体用と、「ホスト・コンポーネントを挿入」ウィザードを使用する個別変換用があります。ただし、必ずしもすべてのコンポーネントおよびウィジェットに、カスタマイズ可能な設定が存在するわけではありません。

注: 縦線 (|) 文字は OR を暗黙指定します。この文字は多くのコンポーネントおよびウィジェット設定で有効です。例えば、ファンクション・キー・コンポーネントには、開始区切り文字と呼ばれる設定があります。ここでは開始区切り文字の値に PF|F を指定できます。これは、HATS が、ファンクション・キーを検索するときにホスト画面上で、PFxxxx または Fxxxx を探すことを意味します。

ホスト・コンポーネントの設定

ホスト・コンポーネントの設定では、そのコンポーネントがホスト画面で認識される方法を指定します。

中には、複雑な設定のホスト・コンポーネントもあります。例えば、いくつかの設定は、ファンクション・キーや選択リストを認識するために使用されます。これらの設定については、各ホスト・コンポーネントの箇所で説明します。

以降のセクションでは、HATS のホスト・コンポーネントについて説明します。

© Copyright IBM Corp. 2003, 2015

コマンド行

コマンド行コンポーネントは、入力フィールドの前に特定の文字ストリング (トークン) を持つ入力フィールドを認識します。

認識されたコマンド行は、以下のウィジェットによるレンダリングが可能になります。

- コンボ RCP のみ
- ドロップダウン (データ入力)
- リスト
- ポップアップ
- ラジオ・ボタン (データ入力)
- テキスト入力
- コンボ・ボックス (Dojo) Web のみ
- フィルター操作選択 (Dojo) Web のみ
- テキスト・ボックス (Dojo) Web のみ

次の図に、ホスト画面上のコマンド行コンポーネントの表示例を示します。

図 12. コマンド行コンポーネントの例

1. フィールドの表題 (抽出)

以下の設定が、このコンポーネントに対して構成できます。

トークン
入力フィールドに先行することが必要な文字列

フィールドの表題を抽出
選択した入力フィールドの使用可能な表題を抽出する場合は、このチェック・ボックスを選択します。

表題のスペースをトリム (Trim spaces on caption)
詳細については、234 ページの『入力フィールド』を参照してください。

選択領域にクリップ
詳細については、234 ページの『入力フィールド』を参照してください。

ダイアログ

ダイアログ・コンポーネントを使用して、デフォルトのレンダリングによるホスト画面上のモーダル (ポップアップ) フレームの認識を支援できます。

ダイアログ・コンポーネントの概念とは、モーダル・ダイアログを使用して作業していることを前提として、境界の記述に対応する、カーソルを囲む長方形の最も小

さな領域を検出することです。モーダル・ダイアログは、開いている間、入力フォーカスを維持します。ユーザーは、別のウィンドウに切り替えるには、ダイアログを閉じるか、またはダイアログにすべてを入力する必要があります。

データを入力してモーダル・ダイアログを完了する際に、HATS ではカーソルがダイアログの境界線内に位置していることを前提とします。HATS は、「**垂直境界文字**」設定で定義されている有効な垂直境界文字を、画面上のカーソル文字の座標から開始し、左側へ検索します。有効な左側の垂直境界文字が検出されると、HATS は同一カーソル座標へ戻り、右側に向かって有効垂直境界文字を検索します。カーソル座標の左右両側で有効垂直境界文字が検出されると、HATS はこの 2 つの文字の間の距離を計測し、有効な長さであることを確認します。

HATS はまた、垂直境界文字の検索と同様の方法で、「**水平境界文字**」設定で定義されている上下の水平境界文字も検索します。

ダイアログ・ウィンドウは長方形であるため、HATS では左右の境界線の列と、上下の境界線の行を使用して、列がモーダル・ダイアログの垂直条件に対応しているかどうか、および行がモーダル・ダイアログの水平条件に対応しているかどうかを判別します。「**最小行数**」属性が満たされていることを前提として、長方形の外側の行と列が位置合わせされると、ダイアログ・コンポーネントはそのウィンドウを一致として宣言し、認識される領域がレンダリングされます。

このコンポーネントを構成するには、HATS プロジェクトの「**プロジェクト設定**」を表示し、「**レンダリング**」タブを選択して「**デフォルト・レンダリング**」をクリックします。

注: ダイアログ・コンポーネントは、デフォルトのレンダリングでのみ使用可能です。

ダイアログ・コンポーネントは、ダイアログ・ウィジェットを使用してレンダリングされます。

次の図に、ホスト画面上のダイアログ・コンポーネントの表示例を示します。

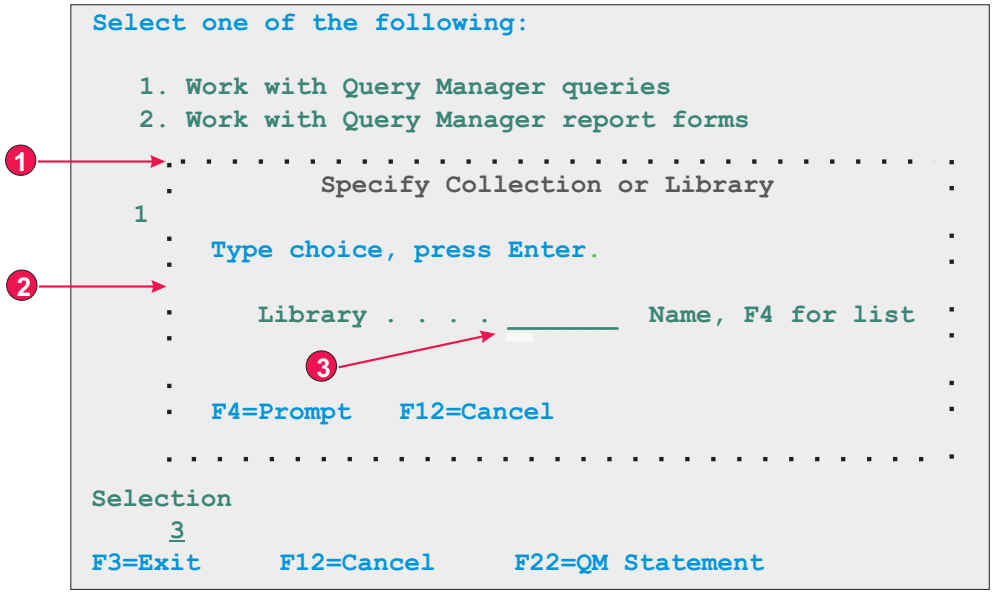


図 13. ダイアログ・コンポーネントの例

1. 水平境界文字
2. 垂直境界文字
3. カーソルをダイアログ内に配置

以下の設定が、このコンポーネントに対して構成できます。

水平境界文字
ダイアログの上下の境界として使用する文字を指定します。デフォルトでは、水平境界文字として保護反転表示文字、ピリオド (.)、ハイフン (-)、アンダースコアー (_) のいずれかを使用できます。複数の文字を使用する場合は、各文字を縦線 (|) で区切ります。これらの文字は、左から右へ順に評価されます。

垂直境界文字
ダイアログの左右の境界として使用する文字を指定します。デフォルトでは、垂直境界文字として保護反転表示文字、コロン (:)、縦線 (|) のいずれかを使用できます。複数の文字を使用する場合は、各文字を縦線 (|) で区切ります。これらの文字は、左から右へ順に評価されます。

注: 縦線 (|) は複数の文字を指定するときの分離文字として使用されるため、垂直境界文字として縦線 (|) を指定するには、&vl.を使用してください。

このコンポーネントの「**水平境界文字**」と「**垂直境界文字**」設定には、文字だけでなく、境界文字のフィールド属性も指定できます。フィールドの属性を指定するときに使用するインディケーターを以下に示します。各フィールドの指定を、中括弧 ({...}) で囲みます。各インディケーターをスペースで区切ります。否定 (NOT) を示すには、インディケーターの前に接頭部として感嘆符 (!) を追加します。インディケーターと感嘆符の間にはスペースを入れないでください。

- 保護フィールド: p

- 隠しフィールド: h
- 数値のみフィールド: no
- 高輝度フィールド: hi
- フィールドの前景色、後続の 16 進文字によって指示される (例: f_7)
- 反転表示フィールド: rv
- 下線フィールド: ul
- 明滅フィールド: b
- 列分離子フィールド: c

例: {!p hi no} は、無保護の強調表示された数値のみのフィールドを示します。

注:

1. 境界文字のフィールド属性を指定するときに、指定部分の直前にデータがある場合は、指定内容はそのデータの修飾子となります。例えば、次のようになります。
   - {p rv} は、任意の保護されている反転表示文字に一致します。
   - :{p rv} は、保護されている 1 つの反転表示コロンに一致します。
   - {p rv}: は、任意の保護されている反転表示文字に一致します。これは、コロンがフィールド指定の後に位置されていることから、コロンには反映されないためです。
2. フィールド属性定義の中のインディケーターの順序は関係ありません。これは、中括弧内のすべてのインディケーターは、認識される境界文字に対し一致する必要があるためです。

最小行数
ダイアログとして認識するために領域に含まれる必要がある最小行数。デフォルトは 3 です。

最小列数
ダイアログとして認識するために領域に含まれる必要がある最小列数。デフォルトは 3 です。

カーソルをダイアログ内に配置
ダイアログとして認識される領域内にカーソルを配置する必要があるかどうかを指定します。これを選択しない場合は、ダイアログとして認識される領域内にカーソルが配置されている必要はありません。モーダル・ダイアログの場合は、予期しない結果が生じることがあります。

ダイアログ外部対話の許可
境界の外側のエリアを使用不可にするかどうかを指定します。

注: ダイアログ境界内部のテキストの置換は、ダイアログ・コンポーネントの設定で定義することはできません。ダイアログ・コンポーネント自体は、ダイアログ・ウィンドウの境界 (まわりのボックス) でしかありません。境界内の内容は、デフォルト・レンダリングのその他のコンポーネントによってレンダリングされます。ダイアログの内容のテキスト置換を実行するために、ダイアログの内容をレンダリングするデフォルト・レンダリング項目のテキスト置換を指定できます。例えば、フィールド・コンポーネントに対しテキスト置換を指定

できます。 ダイアログ・コンポーネントとその他のコンポーネントを組み合わせた正確な表示を取得するには、「**全ページのプレビュー**」をクリックします。

ENPTUI

ENPTUI (拡張非プログラマブル端末ユーザー・インターフェース) を使用すると、5250 フルスクリーン・メニュー方式インターフェースを利用する非プログラマブル端末 (NPT) およびプログラマブル・ワークステーション (PWS) 上で拡張インターフェースを使用できるため、5250 表示データ・ストリーム拡張機能を利用できます。HATS は、グラフィカル・インターフェース構成を含む ENPTUI 機能のサブセットをサポートします。これらの構成は、2 つの HATS コンポーネントである「スクロール・バー」および「選択フィールド」で表されます。

新規 HATS プロジェクトの作成中に ENPTUI サポートを追加するには、HATS ツールバーから「**HATS**」>「**新規**」>「**プロジェクト**」 (または「**ファイル**」>「**新規**」>「**HATS プロジェクト**」) を選択して、「プロジェクトの作成」ウィザードを起動します。「接続設定」パネルが表示されたら、接続タイプとして「**5250**」または「**5250W**」を選択し、「**グラフィカル・インターフェース DDS キーワード (ENPTUI) レンダリング・サポートの追加**」チェック・ボックスを選択してサポートを使用可能にします。

注: このオプションは、モバイル装置に対して最適化されているプロジェクトでは、使用できません。

このチェック・ボックスを選択すると、application.hap ファイル内に設定されているデフォルト・レンダリングに標準 ENPTUI レンダリング項目が含まれます。また、デフォルト・レンダリングかカスタム変換のいずれかが含まれるように、コンポーネントおよびウィジェットのレジストリー (ComponentWidget.xml) に ENPTUI コンポーネントおよびウィジェットが含まれます。また、作成されるデフォルト接続オブジェクトの拡張接続プロパティーに ENPTUI=true が追加されます。この値は、HATS がホストから ENPTUI データを受信するために必要です。

このチェック・ボックスをクリアされたままにすると、「プロジェクトの作成」ウィザードによってコンポーネントおよびウィジェットのレジストリーと application.hap ファイルが編集され、ウィザードが新規プロジェクト内に配置したファイルから ENPTUI レンダリング項目、コンポーネント、およびウィジェットが除去されます。

注: これは、HATS が ENPTUI コンポーネントおよびウィジェットをプロジェクトに追加できる唯一の機会です。作成後に ENPTUI をプロジェクトに追加する場合は、ComponentWidget.xml、application.hap、および main.hco に対するすべての編集を手動で実行する必要があります (比較して必要な変更を見つけることによって実行します)。この方法はお勧めできません。新規プロジェクトを作成して ENPTUI サポートを追加する方法をお勧めします。

ENPTUI コンポーネントとその設定を以下に示します。Page Designer の「**パレット**」ビューの「HATS コンポーネント」セクションには、これらのコンポーネントは表示されません。ENPTUI サポートが上記の手順で使用可能にされている場合に

これらのコンポーネントを操作するには、Rational SDP メニュー・バーから「**HATS ツール**」>「**ホスト・コンポーネントを挿入**」をクリックします。

スクロール・バー・フィールド (ENPTUI)

スクロール・バー・フィールド (ENPTUI) コンポーネントは、5250 データ・ストリームからの ENPTUI スクロール・バー・フィールドの認識を担当します。このコンポーネントは、選択された任意のホスト画面領域を認識し、スクロール・バー (ENPTUI) ウィジェットを使用してレンダリングできます。このコンポーネントには、カスタマイズ可能な設定はありません。

選択フィールド (ENPTUI)

認識された ENPTUI 選択フィールドは、以下のウィジェットによるレンダリングが可能になります。

- ボタン
- チェック・ボックス
- リンク
- ラジオ・ボタン (選択)

以下の設定が、このコンポーネントに対して構成できます。

選択フィールド・タイプ認識 (Selection Field Type Recognition)
コンポーネントの指定された任意のインスタンスで一致させる選択フィールド・タイプ。

プッシュボタン (Push Buttons)
プッシュボタン ENPTUI 選択フィールドを認識させる場合は、このボックスを選択します。

単一選択フィールド (Single Choice Fields)
単一選択 ENPTUI 選択フィールドを認識させる場合は、このボックスを選択します。

複数選択フィールド (Multiple Choice Fields)
複数選択 ENPTUI 選択フィールドを認識させる場合は、このボックスを選択します。

メニュー・バー (Menu Bars)
メニュー・バー ENPTUI 選択フィールドを認識させる場合は、このボックスを選択します。

選択フィールド属性認識 (Selection Field Attribute Recognition)
コンポーネントの指定された任意のインスタンスで一致させる選択フィールド属性。

自動入力を使用できるフィールドの認識 (Recognize fields having auto-enter)
自動入力が使用可能または使用不可のどちらの ENPTUI 選択フィールドを認識させるか (あるいはその両方を認識させるか) をドロップダウン・リストから選択します。

次の図に、ホスト画面上の ENPTUI コンポーネントの表示例を示します。

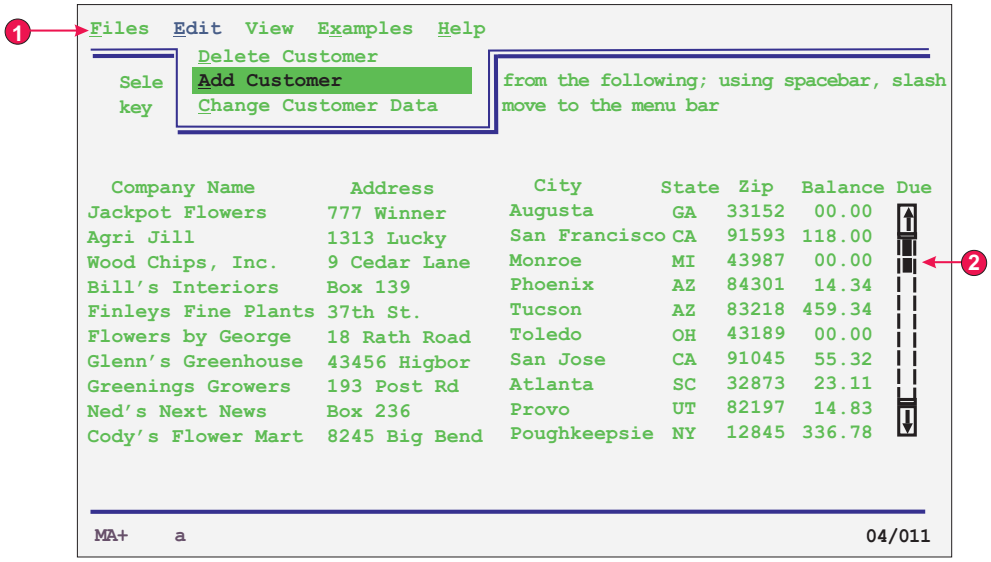


図 14. ENPTUI コンポーネントの例

1. 選択フィールド・コンポーネント
2. スクロール・バー・フィールド・コンポーネント

フィールド

フィールド・コンポーネントは、保護テキストと入力フィールドの認識を担当します。このコンポーネントは、選択された任意のホスト画面領域を認識し、フィールド・ウィジェットを使用してレンダリングできます。

注: フィールド・コンポーネントは、ユーザーによる指定領域のみを含みます。選択領域が、実際のホスト画面フィールドを通り抜けている場合は、選択領域内の部分のみがレンダリングされます。

次の図に、ホスト画面上のフィールド・コンポーネントの表示例を示します。

図 15. フィールド・コンポーネントの例

このコンポーネントには、以下の設定を構成できます。

保護フィールドのサポート文字レベル属性

この設定により、保護フィールドの先頭文字の色と属性でフィールド全体をレンダリングするのではなく、保護フィールドのすべての文字の色と拡張属性を正確にレンダリングできます。

これが選択されていない (デフォルト) 場合は、色などの属性がフィールドの中間で変化する場合でも、フィールドの先頭文字によってフィールド内のすべての文字の属性が決定します。

これが選択されている場合は、一連の類似した文字のセグメントとしてフィールド・コンポーネント・エレメントがレンダリングされ、フィールド内で各種の色および拡張属性 (反転表示など) の情報がサポートされます。例えば、長さ 10 文字のフィールドにおいて、先頭 5 文字が緑色、その後の 5 文字が赤色の場合、このフィールドは 2 つのフィールド・コンポーネント・エレメントにより表現されます。

注: この設定は、保護フィールドにのみ適用されます。入力フィールドには適用されません。

ファンクション・キー

ファンクション・キー・コンポーネントは、ファンクション (PF) キーを認識します。このコンポーネントを使用して、ホスト画面上にあるファンクション・キーを表す静的テキストを、ユーザーがクリック可能なボタンまたはリンクに変換できます。ファンクション・キーは、ホスト・システム間で表示が異なるため、HATS では、そのさまざまな部分を構成する柔軟性が用意されています。例えば、ホスト画面には以下のようなファンクション・キーが含まれます。

F3: 終了 **F4: 戻る** **F5: 進む**

ファンクション・キーは、以下のウィジェットによるレンダリングが可能になります。

- ボタン
- ボタン・テーブル
- ドロップダウン (選択)
- リンク
- ラジオ・ボタン (選択)
- 347 ページの『ツールバー RCP のみ』

次の図に、ホスト画面上のファンクション・キー・コンポーネントの表示例を示します。

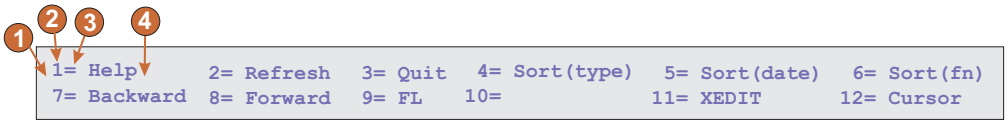


図 16. ファンクション・キー・コンポーネントの例

1. 先行トークンの前のストリング (スペース)
2. 開始区切り文字 (#、数値を示す)
3. 区切り文字 (等号 (=))
4. 説明の後のストリング (スペース)

以下の設定が、このコンポーネントに対して構成できます。

先行トークンの前のストリング

オプション。開始区切り文字に先行する必要のあるストリングです。縦線 (|) 文字で区切って複数の値を指定することができます。認識対象のファンクション・キーの場合、この値が先行する必要があります。

開始区切り文字

必須。ファンクション・キーになる可能性のあるものの開始をマークするストリングです。縦線 (|) 文字で区切って複数の値を指定することができます。いずれの値でもファンクション・キーの開始として認識されます。この値は、1 つの「#」(ポンド記号) を含むことができます。認識中に「1 つの番号」に評価されます。

区切り文字

オプション。開始区切り文字を説明から区切るストリングです。縦線 (|) 文字で区切って複数の値を指定することができます。

説明の後のストリング

オプション。説明の直後に続くストリングです。デフォルトは " " (ダブル・スペース) です。これは、ファンクション・キーの複数語からなる説明の検出を有効にします。

ファンクション・キー・コンポーネント設定を使用する際には、次に説明する例を考慮してください。ホスト画面に次のファンクション・キー説明が示されるとします。

```
F1=Help F2=Open F3=Exit
```

この場合、各説明の間がシングル・スペースで区切られています。「**説明の後のストリング**」設定のデフォルト値は " " (ダブル・スペース) ですが、HATS ではこれは 3 つの個別のファンクション・キーとして認識されます。これは、ファンクション・キー・コンポーネントが**開始区切り文字**、**区切り文字**、**説明の後のストリング**の順に検索するためです。ただし、**説明の後のストリング**が検出される前に新しい**開始区切り文字**が検出されると、これは新規ファンクション・キー説明の開始と解釈されます。

注:

1. HATS ファンクション・キー・コンポーネントは、F# キー、PA キー、Enter およびクリアのみを認識およびレンダリングします。Enter とクリアの認識を使用不可にするには、開始区切り文字の定義から最後の # を除去します。
2. HATS ファンクション・キー・コンポーネントは、ヘルプ、前ページ (ロールアップ)、次ページ (ロールダウン)、システム要求、アテンション、印刷などのその他のホスト・ファンクション・キーを認識しません。これらのその他の機能は、ホスト・キーパッド上で使用できます。「**HATS ツール**」メニュー項目を使用して、これらのホスト・キーを HATS 変換に追加することもできます。
3. ドロップダウン (選択) ウィジェットを使用してファンクション・キー・コンポーネントをデフォルトのレンダリングでレンダリングする場合、プレビュー時および実行時に複数のドロップダウン・ウィジェットがレンダリングされる可能性があります。実行時にこの問題を修正するためには、application.hap ファイルのソースで、SLDropdownWidget が指定された FunctionKeyComponent に対する

<renderingItem ...> タグを見つけます。次に、設定 keepOutputTogether の値を false から true に変更します。プレビューでは、引き続き複数のドロップダウン・ウィジェットが表示されます。

4. 複数のファンクション・キー・コンポーネントをデフォルトのレンダリングで使用していて、変換の中でこれらを別々の行に表示したい場合には、
 タグを使用するか、テーブルにマップして、強制的に移動する必要があります。 これを行うには、「**ソース**」タブで変換を表示して編集します。

HTML DDS キーワード Web のみ

IBM i ディスプレイ・ファイル内で HTML DDS キーワードを使用すると、HTML フラグメントを 5250 データ・ストリームとともに送信できます。この機能は、標準エミュレーション・プログラムでの表示を目的としたものではありません。HTML フラグメントは、ブラウザー内でこの HTML データを表示するように拡張された 5250 Workstation Gateway デバイスおよび HATS などのホスト・アクセス製品にのみ送信されます。

HTML DDS は、ホスト・タイプが 5250 として指定されたときにコンポーネントとして使用できます。このコンポーネントの使用はデフォルトのレンダリングの一部ではないため、必要な場合にはデフォルトのレンダリングまたはカスタム変換に追加する必要があります。

HTML DDS キーワード・コンポーネントは、この HTML データが 5250 データ・ストリーム内に存在する場合に HTML データを認識します。

このコンポーネントは、ラベル・ウィジェットを使用してレンダリングが可能です。

以下の設定が、このコンポーネントに対して構成できます。

フィルターを使用して DDS データを無視 (Ignore DDS data using filter)
DDS データを 5250 データ・ストリーム内で送信されたままの形で認識させる場合、またはこれらの「無視」フィルターで指定されたタグを除いて送信されたままの形で認識させる場合は、このボタンをクリックします。

これらのタグを無視 (Ignore these tags)
無視する任意のタグ (< および > 文字を除く) を入力します。このタグとその終了タグの間のデータは保持されます。例えば、太字タグで囲まれたデータを保持し、タグ自体は保持しない b を指定します。

これらのタグ内のすべての DDS データを無視 (Ignore all DDS data in these tags)
このタグとその終了タグの間のデータを無視する任意のタグ (< および > 文字を除く) を入力します。

フィルターを使用して DDS データを受け入れ (Accept DDS data using filter)
5250 データ・ストリーム内の HTML DDS データからこれらの「受け入

れ」フィルターを通過した HTML タグのみを認識させる場合は、このボタンをクリックします。これらのフィルターを通過しなかった HTML DDS データは無視されます。

これらのタグを受け入れ (Accept these tags)
保持する任意のタグ (< および > 文字を除く) を入力します。このタグとその終了タグの間のデータは無視されます。

これらのタグ内のすべての DDS データを受け入れ (Accept all DDS data in these tags)
このタグとその終了タグの間のデータを保持する任意のタグ (< および > 文字を除く) を入力します。

キーワード HTML を追加すると、ブラウザー形式の受信者のみが解釈する生の HTML データをデータ・ストリームに埋め込むことができます。 HATS でこの機能がサポートされたことにより、ホスト・アプリケーションから HTML DDS キーワード内で送信されたデータを、デフォルトのレンダリングまたはカスタム変換で使用できます。

接続定義の HTML DDS キーワード・サポートを使用可能にするには、以下のステップを実行します。

1. 「**HATS Projects**」ビュー内のメイン HATS 接続をクリックして、「**接続エディター**」を開きます。
2. 「**拡張**」タブを選択し、「**追加**」ボタンをクリックして拡張接続設定を構成します。
3. ドロップダウン・リストから「**名前**」に「**HTMLDDS**」を選択し、「**値**」フィールドに「**true**」と入力します。
4. 「**OK**」をクリックします。

入力フィールド

入力フィールド・コンポーネントは、入力フィールド (無保護フィールド) を認識するためのものです。このコンポーネントは、入力フィールドを認識する以外に、フィールド付随の表題の抽出も行います。入力フィールドの表題を見つけるため、まず、このコンポーネントは、(同じ行上の) 入力フィールドの直前の保護テキストを探します。アルゴリズムは、表題を発見できない場合、保護テキストの入力フィールドの直上を探します。

認識された入力フィールドは、以下のウィジェットによるレンダリングが可能になります。

- カレンダー Web のみ
- チェック・ボックス
- コンボ RCP のみ
- ドロップダウン (データ入力)
- リスト
- ポップアップ
- ラジオ・ボタン (データ入力)

- テキスト入力
- コンボ・ボックス (Dojo) Web のみ
- 日付テキスト・ボックス (Dojo) Web のみ
- フィルター操作選択 (Dojo) Web のみ
- テキスト・ボックス (Dojo) Web のみ
- 検証テキスト・ボックス (Dojo) Web のみ

次の図に、ホスト画面上の入力フィールド・コンポーネントの表示例を示します。

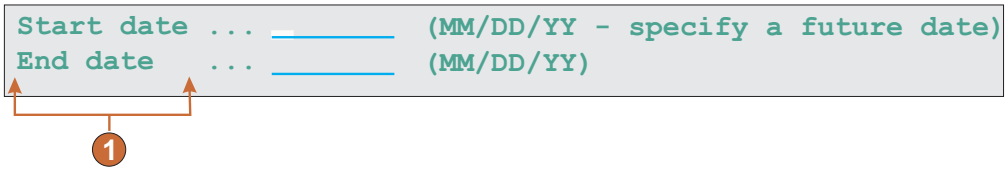


図 17. 入力フィールド・コンポーネントの例

1. フィールドの表題 (コンポーネントから抽出)

以下の設定が、このコンポーネントに対して構成できます。

フィールドの表題を抽出
これを選択した場合、認識された入力フィールドの表題が抽出されます (このアルゴリズムの詳細については、上記の説明を参照してください)。ここで抽出された表題は、ウィジェットが使用できます。

表題を選択領域に制限
選択した場合は、表題が、画面の選択部分のテキスト内のみに制限されます。

表題の終了を除去
これを選択した場合、「**この後を除去**」設定に指定された値が最初に出現した後 (値も含む) から表題の最後までが除去されます。この設定は、抽出した表題を完全に取り去る場合に有効です。

この後除去 (Strip after)
必須。抽出された表題を除去するために使用するストリング (またはストリングの集合) です。例えば、抽出された表題が 「**Command ==>**」で、この設定値が「**=**」の場合は、最初の「**=**」から後 (「=」も含む) のすべてが除去されます。

次と置き換え
オプション。表題の除去された部分に置き換わるストリングです (該当する場合)。

表題のスペースをトリム (Trim spaces on caption)
これを選択した場合、抽出 (および除去) された表題の両端から空白文字 (スペース、タブなど) が削除されます。

入力フィールドを選択領域にクリップ
これを選択した場合、選択領域内の入力フィールドの一部のみが認識および表示されます。クリアした場合は、選択領域内に入力フィールドの一部が存在すると、入力フィールド全体が認識および表示されます。この設定は、大

きなホスト画面入力フィールドを細分化した複数の Web ページ入力フィールドに分割する必要のある場合に有効です。

ヒント付き入力フィールド

ヒント付き入力フィールド・コンポーネントは、付随するヒント (同一行にある入力フィールドの直後のテキスト) を持つ入力フィールド (無保護フィールド) を認識するためのものです。このコンポーネントは機能的には入力フィールド・コンポーネントと同じですが、入力候補の隣にヒントが存在していなければならない点が異なります。例えば、以下の入力フィールドが存在するホスト画面を考えてみます。

製品コード (Product code): [ ] (Valid codes: A, B, C, D)

この例では、ヒントの開始部分が「**Valid codes:**」、ヒントの最後が閉じ括弧「**[)]**」、分離文字がコンマ「**(,)**」、先行トークンのタイプが「**None**」です。

認識された入力フィールドは、以下のウィジェットによるレンダリングが可能になります。

- コンボ RCP のみ
- ドロップダウン (データ入力)
- リスト
- ポップアップ
- ラジオ・ボタン (データ入力)
- コンボ・ボックス (Dojo) Web のみ
- フィルター操作選択 (Dojo) Web のみ

この入力フィールドは、有効な値ヒントも使用します。デフォルトでは、この入力フィールドはヒントを認識しないため、ユーザーがコンポーネント設定を変更して、ヒントをレンダリング・ウィジェットに引き渡す必要があります。

次の図に、ホスト画面上のヒント付き入力フィールド・コンポーネントの表示例を示します。

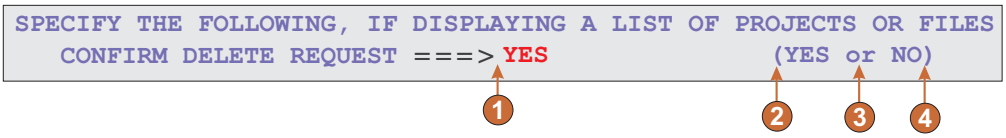


図 18. ヒント付き入力フィールド・コンポーネントの例

1. 区切り文字: 区切り文字の位置。ただし、先行トークンのタイプは「**None**」に設定されていた。
2. ヒントの開始: 開き括弧 (
3. 分離文字: or
4. ヒントの終了: 閉じ括弧)

先行トークンのタイプがマークされていません。この例では、先行トークンのタイプは「**None**」です。ヒントに先行する文字または数字はありません。

以下の設定が、このコンポーネントに対して構成できます。

フィールドの表題を抽出
これを選択した場合、認識された入力フィールドの表題が抽出されます (このアルゴリズムの詳細については、上記の説明を参照してください)。ここで抽出された表題は、ウィジェットが使用できます。

表題を選択領域に制限
選択した場合は、表題が、画面の選択部分のテキスト内のみに制限されます。

表題の終了を除去
これを選択した場合、「**この後を除去**」設定に指定された値が最初に出現した後 (値も含む) から表題の最後までが除去されます。この設定は、抽出した表題を完全に取り去る場合に有効です。

この後除去 (Strip after)
必須。抽出された表題を除去するために使用するストリング (またはストリングの集合) です。例えば、抽出された表題が 「**Command ==>**」で、この設定値が「**=**」の場合は、最初の「**=**」から後 (「=」も含む) のすべてが除去されます。

次と置き換え
オプション。表題の除去された部分に置き換わるストリングです (該当する場合)。

表題のスペースをトリム (Trim spaces on caption)
これを選択した場合、抽出 (および除去) された表題の両端から空白文字 (スペース、タブなど) が削除されます。

入力フィールドを選択領域にクリップ
これを選択した場合、選択領域内の入力フィールドの一部のみが認識および表示されます。クリアした場合は、選択領域内に入力フィールドの一部が存在すると、入力フィールド全体が認識および表示されます。この設定は、大きなホスト画面入力フィールドを細分化した複数の Web ページ入力フィールドに分割する必要のある場合に有効です。

ヒントの開始
オプション。ヒント集合の開始位置を指定する文字列です。この設定には複数の値を指定できます。値は縦線 (|) 文字で分離します。

ヒントの終了
オプション。ヒント集合の終了位置を指定する文字列です。この設定には複数の値を指定できます。値は縦線 (|) 文字で分離します。

分離文字
必須。ヒント集合で各ヒントを区切る文字列です。この設定には複数の値を指定できます。値は縦線 (|) 文字で分離します。

先行トークンのタイプ
各ヒントのタイプを指定します。例えば、ヒントのセットが「**A=Apple,O=Orange,G=Grape**」である場合、A、O、および G がすべて文字であるため、この値には「**文字**」 (または 「**文字または数字**」) をセットすることになります。区切り文字には「**=**」を設定します。

先行トークンの最大長 (Maximum length of leading token)
先行トークンの最大長を指定します。デフォルトは **4** です。

区切り文字
必須。ヒント説明から先行トークンを区切る文字列です。この設定は、先行トークンのタイプが「**None**」でない場合に限り適用されます。

最小限必要なヒント
このコンポーネントが、入力フィールドを認識するための必要最小限のヒント数です。この設定は、誤認識を避ける場合に有効です。

項目の選択

項目選択コンポーネントは、項目リストを含む画面を認識します。このような画面では、項目の横にある文字を選択して対話を行います。

項目選択コンポーネントは、以下のウィジェットによってレンダリングできます。

- チェック・ボックス
- コンボ RCP のみ
- ドロップダウン (データ入力)
- リンク (項目選択)
- リスト
- ポップアップ
- ラジオ・ボタン (項目選択)
- テキスト入力
- コンボ・ボックス (Dojo) Web のみ
- フィルター操作選択 (Dojo) Web のみ

次の図に、ホスト画面上の項目選択コンポーネントの表示例を示します。

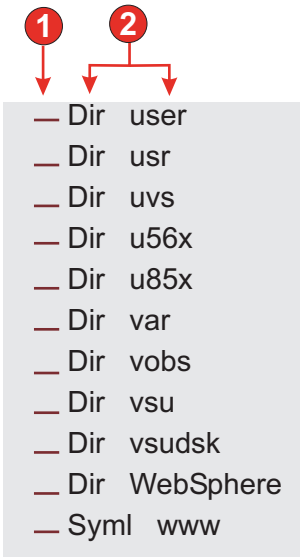


図 19. 項目選択コンポーネントの例

1. 入力フィールド
2. 表題

以下の設定が、このコンポーネントに対して構成できます。

フィールドの表題を抽出
選択した入力フィールドで使用可能な表題を抽出する場合は、このボックスを選択します。

表題の終了を除去
「**フィールドの表題を抽出**」が選択されている場合にのみ使用可能です。表題の終了テキストを削除するには、このボックスを選択します。「**この後を除去**」に削除を開始する位置を指定し、「**置換後**」に削除した表題テキストを置換する文字を指定します。

表題のスペースをトリム (Trim spaces on caption)
「**フィールドの表題を抽出**」が選択されている場合にのみ使用可能です。表題の両端にある余分なスペースをすべて除去するには、このボックスを選択します。

入力フィールドの後の表題 (Captions are after input fields)
「**フィールドの表題を抽出**」が選択されている場合にのみ使用可能です。抽出する表題が入力フィールドの右側にある場合は、このボックスを選択します。表題が入力フィールドの左側にある場合は、このボックスをクリアします。

入力フィールドに表題が必要 (Input fields must have a caption)
ホスト画面上で表題付き入力フィールドのみを認識させる場合は、このボックスを選択します。

入力フィールドを選択領域にクリップ
入力フィールドの選択部分のみをレンダリングする場合は、このボックスを選択します。フィールドの一部が選択されている場合でもフィールド全体をレンダリングする場合は、このボックスをクリアします。

ライト・ペン (アテンション)

ライト・ペン (アテンション) コンポーネントは、3270 ホスト画面上のライト・ペン (セレクター・ペンとも呼ばれる) アテンション・フィールドの認識を担当します。

注: このコンポーネントは、Page Designer の「**パレット**」ビューの「HATS コンポーネント」セクションには表示されません。3270 または 3270E プロジェクトの初期作成時に「**ライト・ペン・レンダリング・サポートの追加**」チェック・ボックスを選択してライトペン・サポートを有効にしている場合、このコンポーネントを使用するには Rational SDP メニュー・バーから「**HATS ツール**」>「**ホスト・コンポーネントを挿入**」をクリックします。

認識されたライト・ペン・アテンション・フィールドは、以下のウィジェットによってレンダリングできます。

- ボタン
- リンク
- ラジオ・ボタン (選択)

指定文字 (フィールドの先頭文字) に従って、レンダリング・ウィジェット (ボタン、リンク、またはラジオ・ボタン) をクリックすると、[enter] または [cursel] アテンション識別子 (AID) がホストに送信されます。指定子がアンパーサンド文字 (&) の場合は、[enter] AID が送信されます。指定子がブランク文字の場合は、[cursel] AID が送信されます。

次の図に、ホスト画面上のライト・ペン (アテンション) コンポーネントの表示例を示します。

図 20. ライト・ペン (アテンション) コンポーネントの例

1. 指定文字 (フィールドの先頭文字)

以下の設定が、このコンポーネントに対して構成できます。

フィールドの残りを消費

ウィジェットが消費およびレンダリングする対象が、フィールド全体またはフィールドの先頭文字のみのいずれであるかを指定します。フィールド全体を消費およびレンダリングすることを指定するには、このチェック・ボックスを選択します。例えば、フィールドの内容がレンダリング・ウィジェット (ボタン、リンク、またはラジオ・ボタン) の表題としてレンダリングされます。ウィジェットにより先頭文字のみをレンダリングすることを指定するには、このチェック・ボックスをクリアします。これは、フィールドの先頭文字より後の部分の色や拡張属性を保持する場合に便利です。デフォルト・レンダリングでは、このチェック・ボックスをクリアすると、フィールドの残

りの部分 (フィールドの先頭文字を除くすべての部分) が他のウィジェット (ほとんどの場合はフィールド・ウィジェット) によりレンダリングされます。デフォルトでは**選択されています**。

ライト・ペン (選択)

ライト・ペン (選択) コンポーネントは、3270 ホスト画面上のライト・ペン (セレクター・ペンとも呼ばれる) 選択フィールドの認識を担当します。

注: このコンポーネントは、Page Designer の「**パレット**」ビューの「HATS コンポーネント」セクションには表示されません。3270 または 3270E プロジェクトの初期作成時に「**ライト・ペン・レンダリング・サポートの追加**」チェック・ボックスを選択してライトペン・サポートを有効にしている場合、このコンポーネントを使用するには Rational SDP メニュー・バーから「**HATS ツール**」>「**ホスト・コンポーネントを挿入**」をクリックします。

認識されたライト・ペン選択フィールドは、チェック・ボックス・ウィジェットによってレンダリングできます。

指定文字 (フィールドの先頭文字) が疑問符 (?) の場合、レンダリングされたチェック・ボックスは選択されません。指定文字がより大記号 (>) の場合は、レンダリング・チェック・ボックスが選択されます。

次の図は、ホスト画面上のライト・ペン (選択) コンポーネントの表示例を示します。

図 21. ライト・ペン (選択) コンポーネントの例

1. 指定文字 (フィールドの先頭文字)

以下の設定が、このコンポーネントに対して構成できます。

フィールドの残りを消費

ウィジェットが消費およびレンダリングする対象が、フィールド全体またはフィールドの先頭文字のみのいずれであるかを指定します。フィールド全体を消費およびレンダリングすることを指定するには、このチェック・ボックスを選択します。例えば、フィールドの内容がレンダリング・ウィジェット (チェック・ボックス) の表題としてレンダリングされます。ウィジェットにより先頭文字のみをレンダリングすることを指定するには、このチェック・ボックスをクリアします。これは、フィールドの先頭文字より後の部分の色や拡張属性を保持する場合に便利です。デフォルト・レンダリングでは、このチェック・ボックスをクリアすると、フィールドの残りの部分 (フィールドの先頭文字を除くすべての部分) が他のウィジェット (ほとんどの場合はフィールド・ウィジェット) によりレンダリングされます。デフォルトが**選択されています**。

選択リスト

選択リスト・コンポーネントは、選択リストの認識を担当します。選択リストはメニューに似ていますが、オプションのリストとして表示される点が違い、以下の例に示すように、各オプションの前に、先行トークンと区切り文字が置かれます。

Option 1. Prepare form
Option 2. Work with forms you submitted
Option 3. Work with forms requiring action

この例では、**Option** が先行トークンに先立つストリング、**1**、**2**、**3** が先行トークン、区切り文字がピリオドとその後のスペース **(.)** です。

リスト項目を選択すると (ウィジェットに基づきクリックまたは選択すると)、その先行トークンが、ターゲットのホストの入力フィールドに置かれます。

認識された選択リストは、以下のウィジェットによるレンダリングが可能になります。

- ボタン
- ボタン・テーブル
- ドロップダウン (選択)
- リンク
- ラジオ・ボタン (選択)
- ツールバー RCP のみ
- コンボ・ボックス (Dojo) Web のみ
- フィルター操作選択 (Dojo) Web のみ

次の図は、ホスト画面上の選択リスト・コンポーネントの表示例を示します。

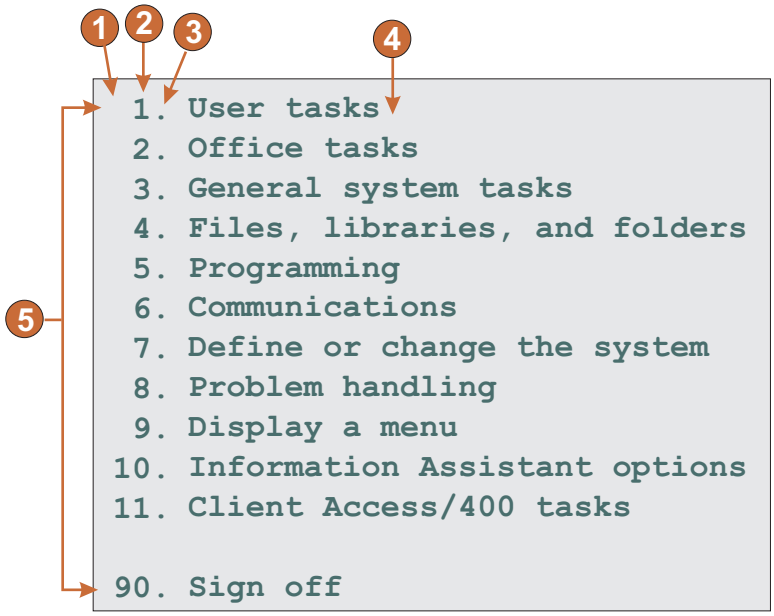


図 22. 選択リスト・コンポーネントの例

1. 先行トークンの前のストリング (スペース)
2. 先行トークンのタイプ (数字)
3. 区切り文字 (ピリオド (.))
4. 終了区切り文字 (スペース)
5. リスト・オプション (または項目)

以下の設定が、このコンポーネントに対して構成できます。

先行トークンの前のストリング
オプション。各リスト項目の先行トークンの前に置かれるストリングです。

先行トークンのタイプ
各リスト項目が、文字、数字、英数字のいずれで始まるかを指定します。

先行トークンの最大長 (Maximum length of leading token)
先行トークンの最大長を指定します。デフォルトは 4 です。

区切り文字
必須。先行トークンと説明の間を区切る文字列です。縦線 (|) 文字で区切って複数の値を指定することができます。

終了区切り文字
オプション。各リスト項目に続く必要のある文字列です。

最小限必要なオプション
選択領域が、選択リストとして認識されるための必要最小限のリスト項目の数です。

項目を個別にグループ化 (Group Items Individually)
項目を個別にグループ化する場合は、このボックスを選択します。

選択ターゲット・フィールド
選択内容を受け取る入力フィールドを指定します。オプション: 「**画面の先**

頭」、「画面の最後」、「前のフィールド」、「次のフィールド」、「カーソル位置」、または「**ユーザー定義**」。

行　選択内容を受け取る入力フィールドの行を指定します。

注: この設定は、「**ユーザー定義**」が選択ターゲットとして選択されている場合に限り適用されます。

列　選択内容を受け取る入力フィールドの列を指定します。

注: この設定は、「**ユーザー定義**」が選択ターゲットとして選択されている場合に限り適用されます。

選択時に自動的に実行依頼

選択されている場合、AID キーは、入力フィールドがブランクでも、自動的にホストに送信されます。クリアにされると、AID キーは、自動的に送信されません。

アクション・キー

オプション。ターゲット入力フィールドがユーザーの選択で更新された後に押すホスト補助キーを指定します。デフォルト値は 「**enter**」 で、エンド・ユーザーが選択を行った後に Enter キーを押すことを示します。

リスト・オプションをソート

これを選択した場合、リスト項目が先行トークンまたは説明に従ってソートされます (他の設定の詳細については、以下を参照してください)。

ソート基準

リスト・オプションの集合をその先行トークンまたは説明のいずれでソートするかを指定します。

ソート順序

リスト・オプションの集合を昇順または降順のいずれでソートするかを指定します。

注: この操作を行うには、Java に組み込まれたロケール依存ソート方式が使用されます。

プレースホルダー・オプション (Placeholder options)

リスト項目間にギャップ (空のスペースなど) が存在する場合にプレースホルダー・リスト項目を埋め込むかどうかを指定します。オプション: 「**実**」、「**なし**」、または「**単一**」。

注: DBCS をサポートするこのコンポーネントの使用の詳細については、547 ページの『選択リスト』を参照してください。

サブファイル

サブファイル・コンポーネントは、5250 サブファイルを認識します。

認識されたサブファイルは、以下のウィジェットによってレンダリングできます。

- サブファイル (チェック・ボックス)
- サブファイル (ドロップダウン)

- サブファイル (ポップアップ)

次の図は、ホスト画面上のサブファイル・コンポーネントの表示例を示します。

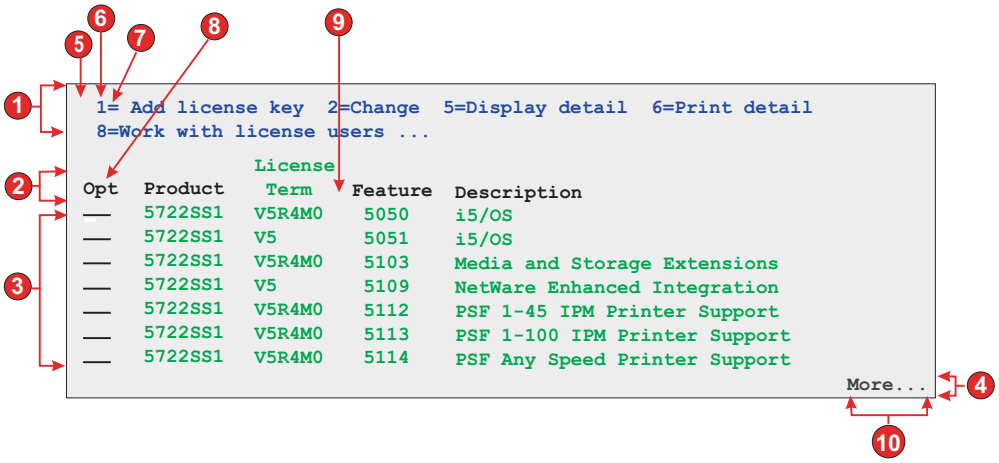


図 23. サブファイル・コンポーネントの例

1. アクション領域
2. ヘッダー領域
3. データ領域
4. マーカー領域

アクション設定

5. 先行トークンの前のストリング (スペース)
6. 先行トークンのタイプ (数字)
7. 区切り文字 (等号 (=))

ヘッダー設定

8. ヘッダーの開始テキスト (オプション)

データ設定

9. 列区切り文字 (スペース)

マーカー設定

10. マーカー・テキスト (続き...)

「サブファイルの設定」を起動するには、以下のステップを行います。

1. 「**HATS Projects**」ビューに移動し、「**プロジェクト設定**」をクリックします。
2. 「**レンダリング**」タブを選択します。
3. 「**レンダリング**」タブで、「**コンポーネント**」ツリーを展開し、「**サブファイル**」コンポーネントを選択します。
4. 右側にある「**設定**」ボタンをクリックして、「サブファイルの設定」ウィンドウを表示します。

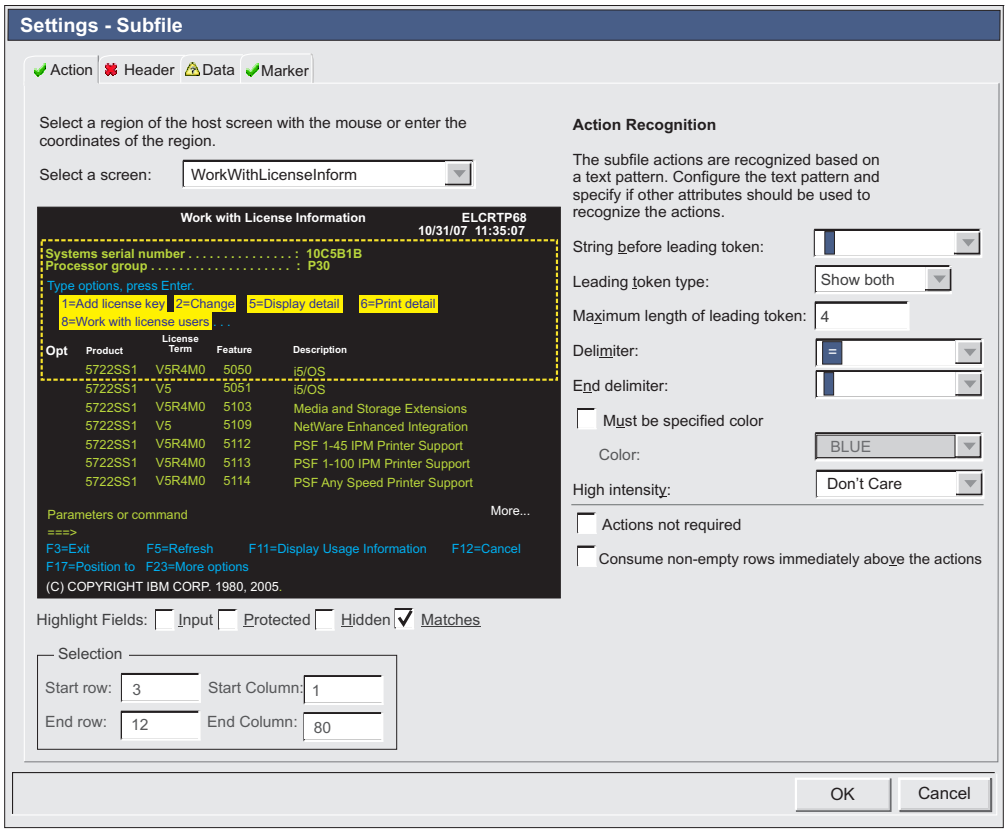


図 24. 「サブファイルの設定」の例

サブファイル・コンポーネント設定は、「アクション」、「ヘッダー」、「データ」、および「マーカー」の 4 つのタブからなります。各タブは、サブファイルの特定のエレメントを認識する基準を定義します。各タブのアイコンは、以下の状態を表します。

- 認識: この状態は、サブファイルのエレメントが、選択された画面で認識されていることを意味します。「**一致**」オプションが選択されている場合、認識されたエレメントは強調表示されます。強調表示の色を変更するには、HATS 設定の「**画面キャプチャーの強調表示時の色**」セクションの「**パターン・マッチング**」設定を使用します。詳しくは、 149 ページの『HATS プリファレンスの使用』を参照してください。
- 非認識: この状態は、サブファイルのエレメントが、認識されていないことを意味します。認識されるまで、画面領域または認識基準を調整し、設定を微調整する必要があります。
- 不明: この状態は、このエレメントの認識が、まだ認識されていない別のエレメントに依存していることを表しています。最初に他のタブの認識されていない設定を微調整してから、このタブに戻ってください。

4 つのタブすべてが認識されたという標識を表示している場合、「**OK**」をクリックします。 ウィジェットのプレビューで認識された結果を表示することができます。

各認識タブの説明を以下に示します。

アクション

使用する画面をドロップダウン・リストから選択します。マウスを使用するかまたは以下のボックス内に領域の座標を入力して、ホスト画面上の「**選択**」エリアの領域を指定することもできます。

開始行 サブファイル・アクションが検索される最初の行です。デフォルトは **3** です。

終了行 サブファイル・アクションが検索される最後の行です。デフォルトは **12** です。

開始列 サブファイル・アクションが検索される最初の列です。デフォルトは **1** です。

終了列 サブファイル・アクションが検索される最後の列です。デフォルトは **80** です。

ホスト画面の特定のフィールドを強調表示するには、「**強調表示するフィールド**」の隣にある各オプションを選択します。入力フィールドが画面上で定義されている位置を表示するには、「**入力**」を選択します。保護フィールドを表示するには、「**保護**」を選択します。隠しフィールドを強調表示するには、「**非表示**」を選択します。 一致するパターンを表示するには、「**一致**」を選択します。入力フィールド、保護フィールド、隠しフィールド、またはパターン・マッチング・フィールドの強調表示の色を変更する方法については、149 ページの『HATS プリファレンスの使用』を参照してください。

アクションの認識
以下の個別設定を選択すると、追加のアクションの認識基準を構成できます。

先行トークンの前のストリング
ストリング。この値は先行トークンの前のストリングを指定します。デフォルト値はスペースです。

先行トークンのタイプ
ストリング。この値は、ホスト・コマンドに対して、ホスト・アプリケーションが使用するトークンのタイプです。「**1=Edit**」という例では、**1** トークンがホスト・コマンドで、そのタイプは数字となります。「**A=Edit**」という例では、**A** トークンがホスト・コマンドで、そのタイプは文字となります。許容値は、「**数字**」、「**文字**」、「**両方を表示**」です。デフォルトは「**両方を表示**」です。

先行トークンの最大長 (Maximum length of leading token)
先行トークンの最大長。デフォルトは **4** です。

区切り文字
ストリング。この値は、サブファイル・アクションで使用されるトークンであり、ホスト・コマンド内の説明からホスト・コマンドを区切ります。「**1=Edit**」という例では、「**1**」がホスト・コマンド、**=** トークンが区切り文字、「**Edit**」 がホスト・コマンドの説明です。許容値は、縦線 (|) で連結された 1 つ以上のストリングです。デフォルトは **=** です。例: **=|** **=** **|-|** **-** **|:**。

終了区切り文字
ストリング。これは、サブファイル・アクション項目の説明の終了を認識するために使用されます。複数の文字を指定する場合は縦線 (|) で区切ります。

指定の色でなければならない
このボックスは前の設定と連動して使用し、サブファイル・アクションのホスト・コマンドが特定のタイプでなければならないこと、およびサブファイル・アクションが特定の色でなければならないことを示します。例えば、サブファイル・アクションは数字に、色は青にすると指定できます。

色 前の設定の説明を参照してください。

高輝度 ストリング。この値は、サブファイル・アクションが高輝度フィールド属性を使用するかどうかを指定します。許容値は、「**はい**」、「**いいえ**」、「**どちらでも可**」です。デフォルトは「**どちらでも可**」です。

アクションは不要 (Actions not required)
サブファイルにアクションがなくてもサブファイルを認識させる場合は、このボックスを選択します。アクションがなくてもサブファイル・ヘッダー、データ、およびマーカーが認識およびレンダリングされます。

アクションのすぐ上にある空でない行を消費
アクションの真上およびすぐ隣にあるテキストを含む画面上の行 (色および/または輝度の認識基準を満たす) をサブファイル・ウィジェットで消費するが表示しない場合、このボックスを選択します。

ヘッダー

マウスを使用するかまたは以下のボックス内に領域の座標を入力して、ホスト画面上の「**選択**」エリアの領域を指定します。

開始行 サブファイル・ヘッダーが検索される最初の行です。デフォルトは **3** です。

終了行 サブファイル・ヘッダーが検索される最後の行です。デフォルトは **16** です。

開始列 サブファイル・ヘッダーが検索される最初の列です。デフォルトは **1** です。

終了列 サブファイル・ヘッダーが検索される最後の列です。デフォルトは **80** です。

ホスト画面の特定のフィールドを強調表示するには、「**強調表示するフィールド**」の隣にある各オプションを選択します。入力フィールドが画面上で定義されている位置を表示するには、「**入力**」を選択します。保護フィールドを表示するには、「**保護**」を選択します。隠しフィールドを強調表示するには、「**非表示**」を選択します。 一致するパターンを表示するには、「**一致**」を選択します。入力フィールド、保護フィールド、隠しフィールド、またはパターン・マッチング・フィールドの強調表示の色を変更する方法については、149 ページの『HATS プリファレンスの使用』を参照してください。

ヘッダーの認識
以下の個別設定を選択すると、追加のヘッダーの認識基準を構成できます。

テキストによって認識

ブール値。この値は、サブファイル・ヘッダーがストリングに基づいて認識されるかどうかを指定します。許容値は、**true** または **false** です。デフォルトは **true** です。

開始 ストリング。サブファイル・ヘッダーの識別に使用できるストリングです。複数行のヘッダーを定義するには、**¥n** を使用します。複数の許容値は縦線 (|) で連結します。より具体的なヘッダーを最初にリストします。例えば、「**Option**」、「**Opt** 」、および 「**Opt**」がすべて考えられるヘッダーである場合、それらは、最も具体的なヘッダーが最初にリストされるような順序で「**Option|Opt |Opt**」とリストされます。デフォルトは **true** です。例: **Description¥nof Record|Option|Opt |?**

指定の色でなければならない

ブール値。この値は「**テキスト**」設定と連動して使用して、サブファイル・ヘッダーが特定のストリングを持ち、特定の色でなければならないことを示します。例えば、サブファイル・ヘッダーは「**Opt.**」という語を含み、色は**青**でなければなりません。許容値は、**true** または **false** です。デフォルトは **false** です。

色 サブファイル・ヘッダーの色を指定します。

高輝度 サブファイル・ヘッダーが、高輝度フィールド属性を使用するデータ・テーブル上部のテキストによって認識できるかどうかを指定するには、このボックスを選択します。

指定の色

データ・テーブルの最初にある特定の色のテキストがサブファイル・ヘッダーを認識できるかどうかを指定するには、このボックスを選択します。

色 サブファイル・ヘッダーの色を指定します。

データ

マウスを使用するかまたは以下のボックス内に領域の座標を入力して、ホスト画面上の「**選択**」エリアの領域を指定します。

開始行 サブファイル・データが検索される最初の行です。デフォルトは **4** です。

終了行 サブファイル・データが検索される最後の行です。デフォルトは **22** です。

開始列 サブファイル・データが検索される最初の列です。デフォルトは **1** です。

終了列 サブファイル・データが検索される最後の列です。デフォルトは **80** です。

ホスト画面の特定のフィールドを強調表示するには、「**強調表示するフィールド**」の隣にある各オプションを選択します。入力フィールドが画面上で定義されている位置を表示するには、「**入力**」を選択します。保護フィールドを表示するには、「**保護**」を選択します。隠しフィールドを強調表示するには、「**非表示**」を選択します。 一致するパターンを表示するには、「**一致**」を選択します。入力フィールド、保護フィールド、隠しフィールド、またはパターン・マッチング・フィールドの強調表示の色を変更する方法については、149 ページの『HATS プリファレンスの使用』を参照してください。

データの認識

以下の個別設定を選択して、追加のデータの認識基準を構成できます。

認識オプション: (Recognition Option:)

この設定では、サブファイル・データ領域の認識方法が指定されます。

フィールド **Web のみ**

サブファイル・データを行単位で認識します。

注: リッチ・クライアント・アプリケーションの場合は、サブファイルを行単位でレンダリングするには、以下のサブファイル・ウィジェット設定を使用します。

- 「**行の代替色の表示**」を選択する
- 「**格子線の表示**」をクリアする
- 「**列見出しのソース**」の「**コンポーネントから**」を選択する
- 「**クラシック見出しスタイルを使用**」を選択する
- 「**フォントのオーバーライド**」を選択し、**モノスペース**・フォントを選択する

テーブル (フィールド) **Web のみ**

サブファイル・データをフィールド情報により認識します。

テーブル (可視)

可視の垂直列区切り文字を使用するテーブルとして表示されるサブファイル・データを認識します。

テーブル

ユーザー定義の列の切れ目を使用するテーブルとして表示されるサブファイル・データを認識します。

列区切り文字:

「**認識オプション:**」が「**テーブル (可視)**」に設定されている場合にのみ使用可能です。テーブルの列としてマークされる対象のホスト画面の列に対して、選択領域のすべての行の同一列に含まれている必要がある文字ストリングです。連続する列が、それぞれ区切り列としてマークされている場合は、最後の列のみが、生成されるテーブルの区切り列として実際にレンダリングされます。デフォルトはスペース文字です。

列の切れ目:

「**認識オプション:**」が「**テーブル**」に設定されている場合にのみ使用可能です。列の切れ目として使用する必要がある列番号をコンマで区切ってリストします。例えば **1,7,22** は、列 1、7、および 22 から新しい列が開始されることを示します。「**デフォルトの復元**」ボタンをクリックすると、「**列の切れ目**」フィールドに、「**テーブル (可視)**」のデフォルト認識値に基づく値が入ります。

注:

1. 列の切れ目を追加または除去するには、マウスを使用してホスト画面ビューで列を選択し、右クリックして「**追加**」または「**除去**」を選択します。ホスト画面ビューでは、列の切れ目は縦の点線として示されます。

2. 列の切れ目によって入力フィールドを分割することはできません。
3. DBCS がサポートされている状態でのこの設定の使用の詳細については、 547 ページの『サブファイル』を参照してください。

アクション入力フィールド開始列をオーバーライド (Override action input field start column)
これを選択した場合は、アクション入力フィールドの開始列をオーバーライドして他の入力フィールドが誤ってドロップダウン・リストとしてレンダリングされないようにできます。

列開始 (Column Start)
アクション入力フィールドの開始列。デフォルト値は **1** です。

アクション入力フィールド長をオーバーライド (Override action input field length)
これを選択した場合は、アクション入力フィールドの長さをオーバーライドして、他の入力フィールドが誤ってドロップダウン・リストとしてレンダリングされないようにできます。

フィールド長
アクション入力フィールドのフィールド長。デフォルト値は **1** です。

データは不要 (Data not required)
サブファイルにデータがない場合でもサブファイルを認識させる場合は、このボックスを選択します。サブファイル・アクションが検出された場合は、ドロップダウン・リストとしてレンダリングされます。

空行を含める
ホスト画面上のテーブルにある空の行を、テーブルの変換時に含める場合は、このボックスを選択します。空の行を破棄する場合は、このボックスをクリアします。

マーカー

マウスを使用するかまたは以下のボックス内に領域の座標を入力して、ホスト画面上の「**選択**」エリアの領域を指定します。

開始行:
サブファイル・マーカーが検索される最初の行です。デフォルトは **4** です。

終了行:
サブファイル・マーカーが検索される最後の行です。デフォルトは **22** です。

開始列 サブファイル・マーカーが検索される最初の列です。デフォルトは **1** です。

終了列 サブファイル・マーカーが検索される最後の列です。デフォルトは **80** です。

ホスト画面の特定のフィールドを強調表示するには、「**強調表示するフィールド**」の隣にある各オプションを選択します。入力フィールドが画面上で定義されている位置を表示するには、「**入力**」を選択します。保護フィールドを表示するには、「**保護**」を選択します。隠しフィールドを強調表示するには、「**非表示**」を選択します。 一致するパターンを表示するには、「**一致**」を選択します。入力フィール

ド、保護フィールド、隠しフィールド、またはパターン・マッチング・フィールドの強調表示の色を変更する方法については、 149 ページの『HATS プリファレンスの使用』を参照してください。

マーカーの認識
以下の個別設定を選択すると、追加のマーカーの認識基準を構成できます。

テキストによって認識
これを選択した場合、指定されたテキスト・ストリングによってサブファイルの終了マーカーを認識できます。

テキスト
ストリング。これは、サブファイル終了マーカーの識別に使用できます。複数の値が使用できます。例: **More...|Bottom|End|+**。デフォルト値は **More...|Bottom|+** です。

高輝度 これを選択した場合、データ・テーブルの下にある高輝度フィールドによってサブファイルの終了マーカーを認識できます。

可視テキストは不要です
これを選択した場合は、高輝度フィールドに空ストリングを挿入できます。

空行 これを選択した場合、データ・テーブルの下にある指定された領域内の空行によってサブファイルの終了マーカーを認識できます。

テーブル

テーブル・コンポーネントは、テーブルを認識します。求められる結果を得るために列の切れ目として特定の列番号を指定する必要がある場合は、テーブル (可視) コンポーネントの代わりにこのコンポーネントを使用できます。

デフォルト・レンダリング設定でテーブル・コンポーネントを使用する場合は注意してください。これは、テーブル・コンポーネントは選択されている領域でほぼすべての画面を認識し、リスト内のこれより下位のレンダリング項目が認識されないためです。

認識されたテーブルは、以下のウィジェットによるレンダリングが可能になります。

- グラフ (水平棒、折れ線、垂直棒)
- テーブル
- 拡張グリッド (Dojo) Web のみ

次の図は、ホスト画面上のテーブル・コンポーネントの表示例を示します。

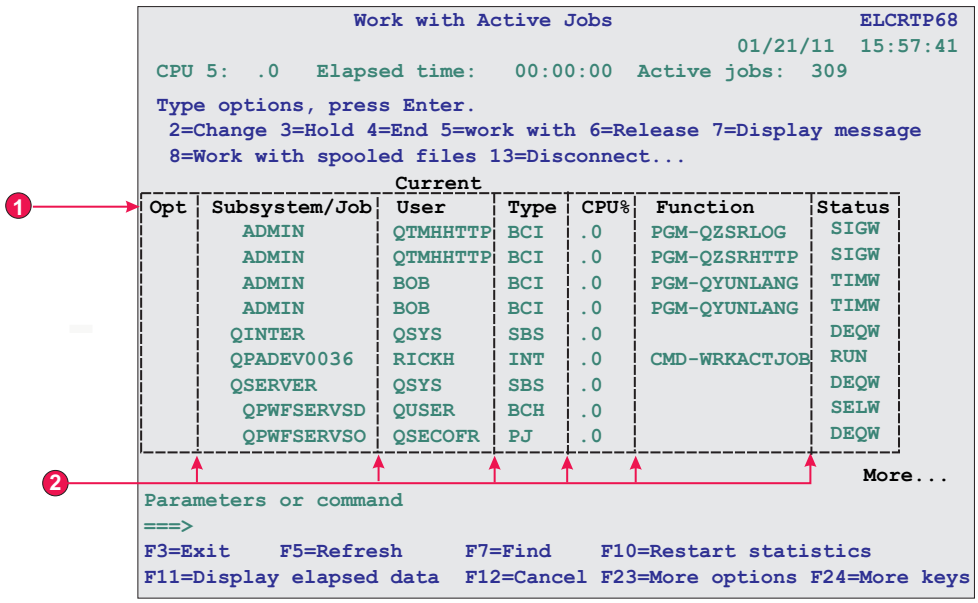


図 25. テーブル・コンポーネントの例

1. タイトル行
2. 列の切れ目

以下の設定が、このコンポーネントに対して構成できます。

列の切れ目:

列の切れ目として使用する必要がある列番号をコンマで区切ってリストします。例えば **1,7,22** は、列 1、7、および 22 から新しい列が開始されることを示します。「**デフォルトの復元**」ボタンをクリックすると、「**列の切れ目**」フィールドに、「**表示されるテーブル**」のデフォルト認識値に基づく値が入ります。

注:

1. 列の切れ目を追加または除去するには、マウスを使用してホスト画面ビューで列を選択し、右クリックして「**列を追加**」または「**列を除去**」を選択します。列の左クリック操作でも、列の切れ目を追加または除去できます。ホスト画面ビューでは、列の切れ目は縦の点線として示されます。
2. 列の切れ目によって入力フィールドを分割することはできません。
3. DBCS がサポートされている状態でのこの設定の使用の詳細については、 547 ページの『テーブル』を参照してください。

空行を含める

これを選択した場合、表示されたテーブルまたはグラフに空行 (表示テキストまたは入力フィールドがない行) が含まれます。

除外するテーブル行

認識されたテーブルから除外するコンマで区切られた行や範囲指定された行セット。この設定は、テーブルまたはグラフ上の特定の複数行のデータを隠

す場合に有効です。この値で表される行は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。

除外するテーブル列
認識されたテーブルから除外するコンマで区切られた論理列や範囲指定された論理列セット。この値で表される列は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。

タイトル行の数
ヘッダーとしてレンダリングする行の領域上部からの行数を指定します。1 よりも大きい値を指定すると、列ごとにヘッダー行が単一行のストリングとしてレンダリングされます。

テーブル (フィールド)

テーブル (フィールド) コンポーネントは、縦方向に位置合わせされたフィールドの論理列を認識します。テーブル (フィールド) として認識される領域の場合、選択領域内に含まれる各フィールド (部分フィールドも含む) の左右の境界線は、その直上のフィールド (問題のフィールドが最初の行でない場合) および直下のフィールド (問題のフィールドが最後の行でない場合) の左右の境界線と同じである必要があります。

認識されたテーブル (フィールド) は、以下のウィジェットによってレンダリングできます。

- グラフ (水平棒、折れ線、垂直棒)
- テーブル
- 拡張グリッド (Dojo) Web のみ

次の図に、ホスト画面上のテーブル (フィールド) コンポーネントの表示例を示します。

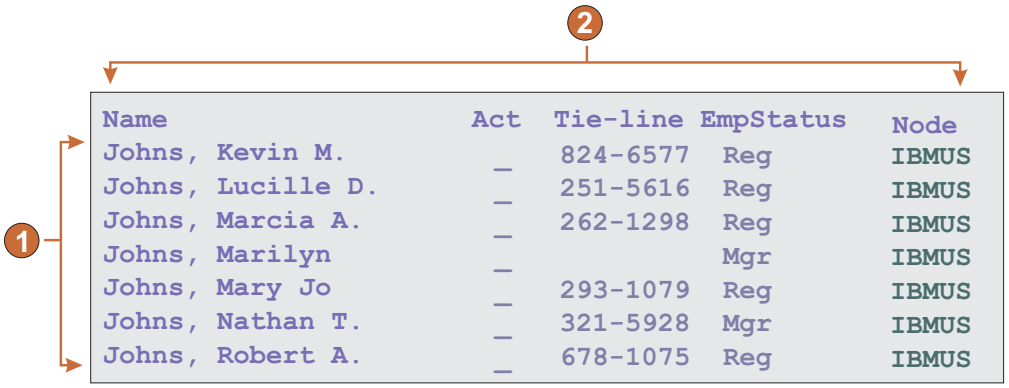


図 26. テーブル (フィールド) コンポーネントの例

1. 行
2. 列

以下の設定が、このコンポーネントに対して構成できます。

空行を含める

これを選択した場合、表示されたテーブルまたはグラフに空行 (表示テキストまたは入力フィールドがない行) が含まれます。

除外する行

認識されたテーブルから除外するコンマで区切られた行や範囲指定された行セット。この設定は、テーブルまたはグラフ上の特定の複数行のデータを隠す場合に有効です。この値で表される行は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3...**、**1,2-4**、**3-5**。

除外する列

認識されたテーブルから除外するコンマで区切られた論理列や範囲指定された論理列セット。この値で表される列は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3...**、**1,2-4**、**3-5**。

最小行数

指定された領域が、テーブルと認識されるために含むべき必要最小限の行数です。

注: 認識された行の数は、指定した行が除外される前に評価されます (『**除外する行**』を参照してください)。

最小列数

指定された領域が、テーブルと認識されるために含むべき必要最小限の (ホスト画面列ではなく) 論理列の数です。

注: 認識された列の数は、指定した列が除外される前に評価されます (『**除外する列**』を参照してください)。

テーブルの上の行から列見出しテキストを抽出

テーブルの上の行にあるテキストをテーブルの列ヘッダーとして使用する場合は、このボックスを選択します。列ヘッダーは保護フィールド内に位置している必要があります。

注: テーブルを作成したホスト画面マップへのアクセス権限がない場合、テーブル (可視) コンポーネントを使用してテーブル領域を認識します。この操作が可能でない場合、「ホスト・コンポーネントを挿入」ウィザードの最初のページの「**入力**」、「**保護**」、および「**非表示**」の各オプションをオンにして、選択したホスト画面領域内のフィールドの区切りを判別しやすくします。

テーブル (可視)

テーブル (可視) コンポーネントは、テーブルを認識します。テーブル (可視) コンポーネントは、基底のホスト画面フィールドが垂直方向に整列されている必要がない点でテーブル (フィールド) コンポーネントと異なります。このコンポーネントは、選択領域で、選択内の各行に列区切り文字を含む列を検索します。経験法則: 領域がテーブルのように示されている場合は、通常、テーブル (可視) コンポーネントがその領域を認識できます。

認識されたテーブルは、以下のウィジェットによるレンダリングが可能になります。

- グラフ (水平棒、折れ線、垂直棒)
- テーブル
- 拡張グリッド (Dojo) Web のみ

次の図に、ホスト画面上のテーブル (可視) コンポーネントの表示例を示します。

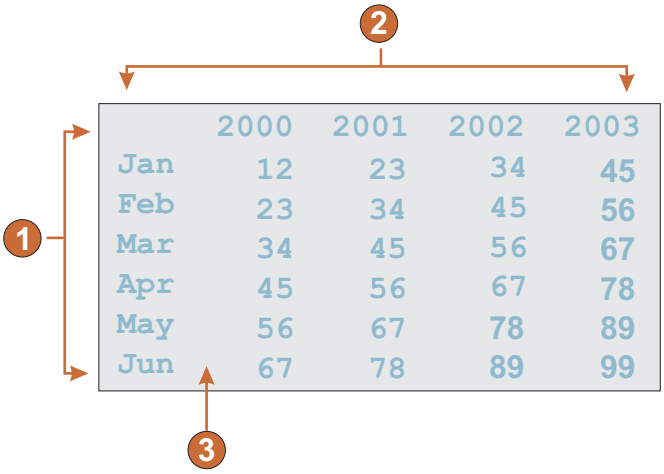


図 27. テーブル (可視) コンポーネントの例

1. 行 (最小行数 = 1)
2. 列 (最小列数 = 1)
3. 列区切り文字 (スペース)

以下の設定が、このコンポーネントに対して構成できます。

列区切り文字
必須。テーブルの列としてマークされているホスト画面列の選択された領域の各行に存在している必要がある文字ストリング。連続する列が、それぞれ区切り列としてマークされている場合は、最後の列のみが、生成されるテーブルの区切り列として実際にレンダリングされます。

空行を含める
これを選択した場合、表示されたテーブルまたはグラフに空行 (表示テキストまたは入力フィールドがない行) が含まれます。

除外する行
認識されたテーブルから除外するコンマで区切られた行や範囲指定された行セット。この設定は、テーブルまたはグラフ上の特定の複数行のデータを隠す場合に有効です。この値で表される行は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。

除外する列
認識されたテーブルから除外するコンマで区切られた論理列や範囲指定された論理列セット。この値で表される列は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。

最小行数
指定された領域が、テーブルと認識されるために含むべき必要最小限の行数です。

注: 認識された行の数は、指定した行が除外される前に評価されます (『**除外する行**』を参照してください)。

最小列数
指定された領域が、テーブルと認識されるために含むべき必要最小限の (ホスト画面列ではなく) 論理列の数です。

注: 認識された列の数は、指定した列が除外される前に評価されます (『**除外する列**』を参照してください)。

テーブルの上の行から列見出しテキストを抽出
テーブルの上の行にあるテキストをテーブルの列ヘッダーとして使用する場合は、このボックスを選択します。列ヘッダーは保護フィールド内に位置している必要があります。

テキスト

テキスト・コンポーネントは、入力フィールド内のテキストを含めて、表示されるテキストの認識を担当します。

このコンポーネントは、ラベル・ウィジェットを使用してレンダリングが可能です。

次の図は、ホスト画面上のテキスト・コンポーネントの表示例を示します。

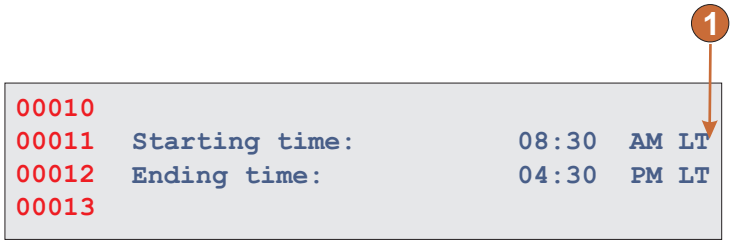


図 28. テキスト・コンポーネントの例

1. 改行 (保存)

以下の設定が、このコンポーネントに対して構成できます。

改行を保存
これを選択した場合、選択した複数の行が、結果の Web ページ上で複数の行としてレンダリングされます。クリアした場合は、各行のテキストが単一のストリングに連結されます。

URL

URL コンポーネントは、リンクの認識を担当します。このコンポーネントは、保護された非隠しフィールド内のみを検索します。 このコンポーネントは、リンク・ウィジェットを使用してレンダリングが可能です。

次の図は、ホスト画面上の URL コンポーネントの表示例を示します。

図 29. URL コンポーネントの例

1. 開始標識 (http://)

以下の設定が、このコンポーネントに対して構成できます。

開始標識
ホスト画面上で URL の開始を認識する 1 つ以上の文字ストリングを指定します。

ウィジェット設定

ホスト・コンポーネントをカスタマイズする場合、ホスト・コンポーネントを認識する方法を指定します。ウィジェットをカスタマイズするのは、ウィジェットが Web ページ上で表示される方法を指定するためです。

以降のセクションでは、カスタマイズ可能なウィジェットの設定について説明します。

ウィジェット設定ダイアログでウィジェット設定を設定または変更すると、ユーザーが「OK」をクリックして変更内容をコミットする前に、ウィジェット・プレビュー領域が自動的に更新され、行った変更の結果が表示されます。

ボタン

ボタン・ウィジェットは、HTML ボタンのレンダリングを担当します。HTML ボタンの 1 つをクリックすると、普通、ホスト指定アクションが実行されます (例えば、補助キーがホストに送信されるか、入力フィールドが更新されます)。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- ファンクション・キー
- ライト・ペン (アテンション)
- 選択リスト
- 選択フィールド (ENPTUI)

次の図に、変換においてファンクション・キー・コンポーネントの例のデータを入力として使用するボタン・ウィジェットの表示例を示します。

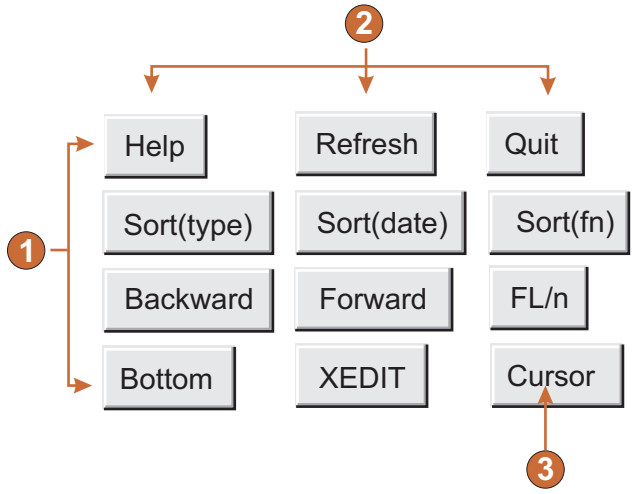


図 30. ボタン・ウィジェットの例

1. 行
2. 列 (1 行あたり 3)
3. 表題のタイプ (説明を表示)

以下の設定が、このウィジェットに対して構成できます。

表題のタイプ
: 各ボタンの表題を決定する方法を指定します。コンポーネントから先行トークンの値および説明が取り込まれます。つまり、ボタンの表題上に表示される項目を選択できます。例えば、ホスト画面に「**4. メール**」というメニュー項目が存在する場合、表題表示として「**4**」、「**メール**」、または「**4. メール**」を選択できます。

表題のスペースをトリム (Trim spaces on caption)
: これを選択した場合、表題の両端から空白 (余分なスペース) が除去されます。

レイアウト
: HTML ページでのボタンの配置方法を指定します。オプション: 「**テーブル**」、「**分離**」。

  注:

  1. この設定はファンクション・キー・コンポーネントが使用されている場合、デフォルト・レンダリングでは適用されません。
  2. ウィジェットのプレビューは必ずしも全ページのプレビューと一致するとは限りません。これはウィジェットのプレビューが単にコンポーネントの出力をレンダリングしているだけであるために起こります。ウィジェットのプレビューは、全ページのプレビューで行われているように、レンダリングしたときに画面の配置を保存しようとはしません。

1 行あたりの列数
: 次の行に折り返されるまでに、水平方向に表示されるボタンの数です。この設定を微調整して、ボタンを垂直方向に (この値が 1 の場

合) または水平方向に (この値が、レンダリング対象ボタンの数以上の場合) 表示することができます。

分離文字

HTML ページ上でレンダリングされたボタンを区切るために使用する分離文字を指定します。ドロップダウン・リストから選択するか入力フィールドに手動で入力するかを選択できます。

前景色を使用可能にする **Web のみ**

これを選択した場合、ボタン・テキストが、ホスト画面から抽出された色を使用して、コンポーネントによってレンダリングされます (ファンクション・キー・コンポーネントの使用時に限り適用可能)。

色情報は、ファンクション・キーのいずれかの部分を含む最初のホスト・フィールドに対して抽出されます。 ファンクション・キーが、F12= および Exit などの 2 つのフィールドに分割された場合、F12= を含むフィールドの色情報が抽出されます。

ボタンのスタイル・クラス **Web のみ**

オプション。生成される各ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**

オプション。オーバーライドする任意の CSS プロパティー。例えば、このフィールドに「**font-color: red; font-size: 18pt;**」と指定して、このウィジェットのフォントの色およびサイズを変更できます。入力したプロパティーは、このウィジェットの各エレメントに適用されます。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド **RCP のみ**

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

ボタン・テーブル

ボタン・テーブル・ウィジェットは、垂直方向に並んでいる HTML ボタンのテーブルのレンダリングを担当します。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- ファンクション・キー
- 選択リスト

次の図に、変換においてファンクション・キー・コンポーネントの例のデータを入力として使用するボタン・テーブル・ウィジェットの表示例を示します。

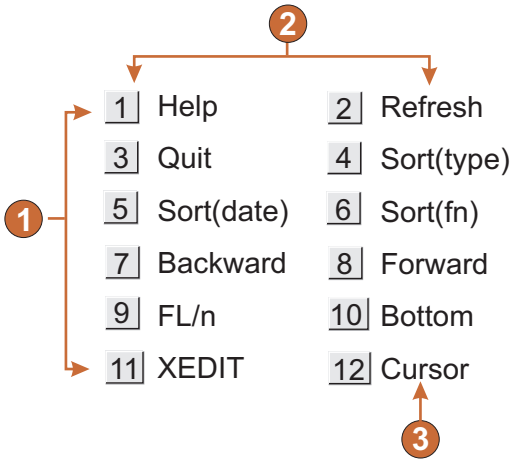


図 31. ボタン・テーブル・ウィジェットの例

1. 行
2. 1 行あたりの列数 (2)
3. 表題

以下の設定が、このウィジェットに対して構成できます。

1 行あたりの列数
次の行に折り返されるまでに、水平方向に表示されるボタンの数です。

注:

1. この設定はファンクション・キー・コンポーネントが使用されている場合、デフォルト・レンダリングでは適用されません。
2. ウィジェットのプレビューは必ずしも全ページのプレビューと一致するとは限りません。これはウィジェットのプレビューが単にコンポーネントの出力をレンダリングしているだけであるために起こります。ウィジェットのプレビューは、全ページのプレビューで行われているように、レンダリングしたときに画面の配置を保存しようとはしません。

ボタンのスタイル・クラス **Web のみ**
オプション。生成される各ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

表題のスタイル・クラス **Web のみ**
オプション。 (生成されるボタンの右側の) 各項目の説明に関連する CSS スタイル・クラスです。デフォルト値は **HATSCAPTION** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス **Web のみ**
オプション。生成されるテーブルに関連する CSS スタイル・クラスです。

HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルトは **HATSTABLE** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**

オプション。オーバーライドする任意の CSS プロパティー。例えば、このフィールドに「**font-color: red; font-size: 18pt;**」と指定して、このウィジェットのフォントの色およびサイズを変更できます。入力したプロパティーは、このウィジェットの各エレメントに適用されます。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド **RCP のみ**

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド **RCP のみ**

前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

カレンダー Web のみ

カレンダー・ウィジェットは、関連のランチャー・ボタン、リンク、またはイメージを持つ入力フィールドのレンダリングを担当します。このランチャーでは、日付を選択するのに使用できるポップアップ・カレンダー日付ピッカーが開きます。選択された日付は、関連入力フィールドに挿入されます。

注:

1. Rational SDP に組み込まれた Web ブラウザーを使用して WebSphere テスト環境でプロジェクトをテストする場合は、予想よりも大きなブラウザー・ウィンドウ内にカレンダー・ウィジェットが表示され、追加の不要なコントロールが表示されることがあります。外部ブラウザーでアプリケーションを実行する場合またはデプロイ済みアプリケーションを実行する場合には、この問題は発生しません。
2. プロキシー・サーバーを介して HATS アプリケーションがアクセスされたときにポップアップ・カレンダーおよびイメージ・ランチャーを表示するには、プロキシー・サーバーを使用するよう HATS アプリケーションを構成する必要があります。手順については、 44 ページの『プロキシー・サーバーを使用するための HATS アプリケーションの構成』を参照してください。
3. このウィジェットを使用する際の DBCS に関する考慮事項については、 548 ページの『SBCS での最大長の除去』を参照してください。

次の図は、「**インライン・カレンダーを使用 (Use inline calendar)**」設定が選択されているときに、変換でカレンダー・ウィジェットとその日付ピッカーがどのよう

に表示されるかを示しています。

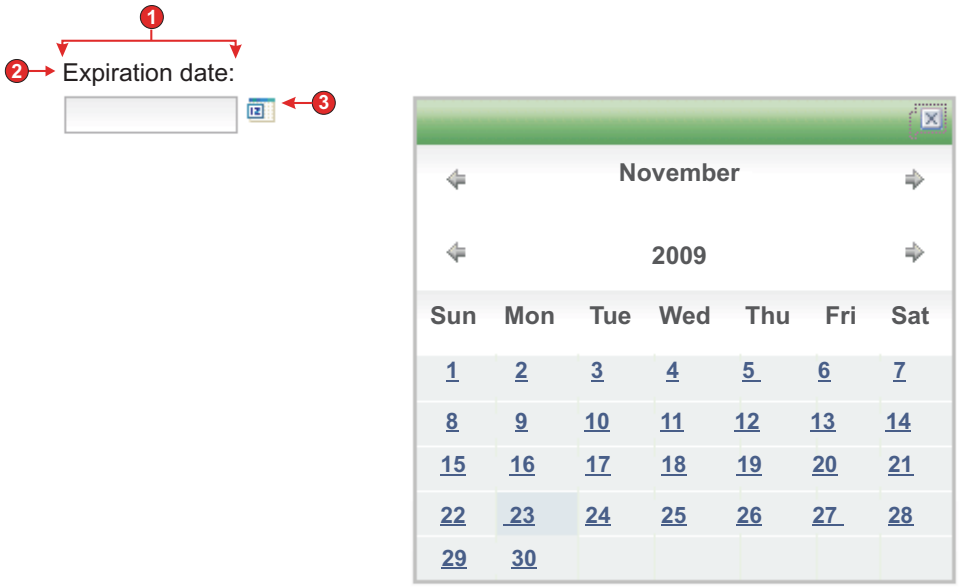


図 32. インライン日付ピッカーを使用したカレンダー・ウィジェットの例

次の図は、「**インライン・カレンダーを使用 (Use inline calendar)**」設定がクリアされているときに、変換でカレンダー・ウィジェットとその日付ピッカーがどのように表示されるかを示しています。

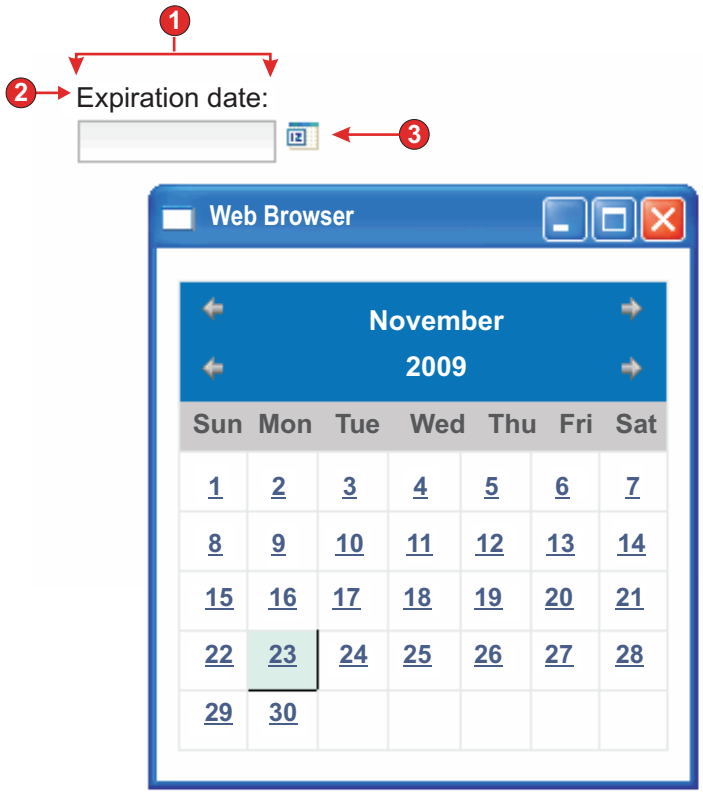


図 33. ウィンドウ日付ピッカーを使用したカレンダー・ウィジェットの例

1. 1 行あたりの列数 (1)
2. 表題 (コンポーネントから)
3. ランチャー・タイプ (イメージ)

注: これらの例では、ランチャーはイメージであるため、ランチャーに対する表題はありません。ランチャーがボタンまたはリンクである場合は、ランチャーの表題がボタンまたはリンクに表示されます。

以下の設定が、このウィジェットに対して構成できます。

パターン

必須。ユーザーがカレンダー・ウィジェットから日付を選択すると、選択された日付をホスト・アプリケーションの入力フィールドで予期されるフォーマットに正しくマップするようにフォーマット設定する必要があります。この設定は、選択された日付を正しくフォーマット設定するために HATS が

使用するパターンを指定します。パターン内の記号の意味については、http://java.sun.com/j2se/1.3/docs/api/java/text/SimpleDateFormat.html を参照してください。

パターンを入力して保存すると、HATS Toolkit は正しい記号が使用されていることを確認するために以下の変換を実行します。

- 大文字の D から小文字の d
- 大文字の Y から小文字の y
- 小文字の m から大文字の M
- 小文字の e から大文字の E

注: HATS Toolkit は、常にパターンを大文字で表示します。パターン・シンボルの実際の大/小文字を参照するには、application.hap ファイルのソースで com.ibm.hats.transform.widgets.CalendarWidget クラスのパターン設定を参照してください。

パターン要素をチルド (~) で区切ることによって、選択されている日付を複数の連続ホスト入力フィールドに分散させることができます。例えば、ホスト画面上に以下の入力フィールドが存在するものとします。

Month: [ ] Day: [ ] Year: [ ]

MM~DD~YYYY というパターンを指定すると、単一のランチャーが作成されます。ユーザーが日付を選択すると、日付は 3 つの入力フィールドに挿入されます。月 (2 桁) が最初の入力フィールド、日 (2 桁) が 2 番目の入力フィールド、および年 (4 桁) が 3 番目の入力フィールドに挿入されます。

注: 複数のホスト入力フィールドにわたるパターンを指定する補助機能を使用するには、「**ビルド**」ボタンをクリックします。

パターン・ロケール

パターンのロケールを指定します。この設定は、曜日の名前 (例: 水曜日) を出力するパターン・セグメントを入力した場合に便利です。オプション: 「**サーバーのロケールを使用**」、「**ブラウザーのロケールを使用**」、または「**指定のロケールを使用**」。

ロケール (Locale)

「**指定のロケールを使用**」を指定した場合は、ユーザーによって指定された日付をフォーマット設定するロケールを選択します。

選択可能な最も古い日付を制限

これを選択した場合、指定される値は、ユーザーがポップアップ・カレンダーから選択可能な最も古い日付となります。

注:

1. すべてのロケールで、日付パターンは MM/DD/YYYY です。
2. この設定では、関連する入力フィールドにユーザーが古い日付を手動で直接入力する操作は防止されません。

選択可能な最も新しい日付を制限

これを選択した場合、指定される値は、ユーザーがポップアップ・カレンダーから選択可能な最新の日付となります。

注:

1. すべてのロケールで、日付パターンは MM/DD/YYYY です。
2. この設定では、関連する入力フィールドにユーザーが新しい日付を手動で直接入力する操作は防止されません。

デフォルト値

オプション。このフィールドの値は、ホスト・アプリケーションで関連の入力フィールドに、有効な日付 (指定された日付制限の間の日付) が事前に設定されなかった場合、ポップアップ・カレンダーで選択される初期日付です。また、ユーザーがポップアップ・カレンダー・コントロールを選択する前に入力した日付が誤っている場合 (範囲外の日付など) にも使用されます。このフィールドと関連するホスト入力フィールドとの間の相互作用は、次のように処理されます。

- ホスト・アプリケーションにより入力フィールドにゼロまたは不適切な日付形式が事前に設定されると、このホスト・フィールドのデータはデフォルト値によって上書きされません。関連の入力フィールドを更新するには、ユーザーが手動で日付を入力するか、またはポップアップ・カレンダーから日付を選択する必要があります。
- デフォルト値が指定されていないか、またはデフォルト値のフォーマットが誤っている場合、本日の日付が最初にポップアップ・カレンダーで選択されます。
- 関連の入力フィールドにデータを事前に設定するには、変換を適用する前に、画面カスタマイズ・イベントに「**データを挿入**」アクションを追加します。

注: すべてのロケールで、日付パターンは MM/DD/YYYY です。

表題ソース (Caption source)

生成される入力フィールドの表題を決定する方法を指定します。オプション: 「**コンポーネントから**」 (コンポーネントから取り出した表題を使用) または「**カスタム**」(指定された表題を使用)。

カスタム表題

オプション。入力フィールドの表題を指定します。

注: 入力フィールドの表題を生成しない場合は、この設定をブランクのままにしてください。

1 行あたりの列数

レンダリングした Web ページの各行に表示するこのウィジェットのインスタンスの数を指定します。

注:

1. この設定はデフォルト・レンダリングでは適用されません。
2. ウィジェットのプレビューは必ずしも全ページのプレビューと一致するとは限りません。これはウィジェットのプレビューが単にコンポーネントの出力をレンダリングしているだけであるために起こります。ウィジェットのプレビューは、全ページのプレビューで行われているように、レンダリングしたときに画面の配置を保存しようとはしません。

ランチャー・タイプ

ランチャーのスタイルを指定します。ランチャーを使用して、カレンダー日付ピッカーを開きます。オプション: 「**ボタン**」、「**リンク**」、または「**イメージ**」。

表題　ランチャー・ボタンまたはリンクの表題。

ボタンのスタイル・クラス

オプション。生成されるランチャー・ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

リンクのスタイル・クラス

オプション。生成されるランチャー・リンクに関連する CSS スタイル・クラスです。HTML リンク・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSLINK** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

注: Web ブラウザーに応じて、「**リンク**」スタイルまたは「**イメージ**」スタイルのいずれかが選択されている場合、タブ・キーを使用したこのランチャーへのタブ移動に制限が存在する可能性があります。サポートされる Web ブラウザーのリストおよび制限については、「Rational Host Access Transformation Services: System Requirements」(http://www.ibm.com/support/docview.wss?uid=swg27011794) および「Host Access Transformation Services 9.5 - Known issues and workarounds」(http://www.ibm.com/support/docview.wss?uid=swg27046998) を参照してください。

インライン・カレンダーを使用

選択されている場合、カレンダー日付ピッカーが起動され、現在の Web ページ内に表示されます。クリアされている場合、日付ピッカーは新規ブラウザー・ウィンドウに表示されます。新規に作成されるプロジェクトでは、デフォルトで選択されています。HATS V7.5.1 より前のリリースからマイグレーションされるプロジェクトの場合、デフォルトでクリアされています。

注:

1. インライン・カレンダー日付ピッカーの使用は、ポップアップ・ブロッカーにより新規ブラウザー・ウィンドウで日付ピッカーを表示できない場合に役立ちます。
2. インライン・カレンダー日付ピッカーの使用は、モバイル装置用の HATS アプリケーションの場合はサポートされません。

読み取り専用

これを選択した場合、レンダリングされた入力フィールドは読み取り専用になります。読み取り専用の入力フィールドは通常の入力フィールドのように表示されますが、ユーザーはこのフィールドの内容を変更できません。これは、無保護フィールドの内容をユーザーに対して表示するが、ユーザーがフィールドの内容を変更できないようにする場合に便利です。

入力フィールドの下線文字を除去 (Strip underscores on input field)
テキストをレンダリングするときにテキストから下線文字を除去する場合は、このボックスを選択します。

入力フィールドのスペースをトリム
入力フィールドから先行および後続のスペースをトリムする場合は、このボックスを選択します。

前景色を使用可能にする
これを選択した場合、ホスト画面の前景色がレンダリングされます。

色はその色を表す CSS スタイルシート・クラスにマップされています。例えば、ホスト画面フィールドが **RED** とマークされている場合、フィールド・ウィジェットは、そのフィールド用に生成された HTML を、クラス名属性が **HRED** に設定されているタグで囲みます。この処理によって、ユーザーは、生成される Web ページ上にホスト画面の色を再マップすることができます。

注: blacktheme.css は、ホスト画面フィールドの前景色を直接マップします (例えば、ホスト画面上の赤色のテキストは、生成された Web ページ上で赤色のテキストとして表示されます)。ただし、monochrome.css などの他のスタイルシート・ファイルは、フィールドの色を別の方法でマップし、一貫性のある、最新のスタイルを作成しようとします。

拡張属性を使用可能にする
これを選択した場合、拡張フィールド属性 (明滅、反転表示、下線、および列分離子) が表示されます。また、3270 Web アプリケーションの場合、拡張フィールドの色もマップされます (詳細については、「**前景色を使用可能にする**」の設定を参照してください)。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

明滅スタイル
ホスト画面における明滅テキストのレンダリングに使用する CSS スタイル設定。

反転表示スタイル (Reverse video style)
ホスト画面における反転表示テキストのレンダリングに使用する CSS スタイル設定。

下線スタイル (Underline style)
ホスト画面における下線付きテキストのレンダリングに使用する CSS スタイル設定。

列分離子スタイル (Column separator style)
ホスト画面における列分離子のレンダリングに使用する CSS スタイル設定。

入力フィールドのスタイル・クラス
オプション。生成される入力フィールドに関連する CSS スタイル・クラスです。HTML 入力タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSINPUT** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

表題のスタイル・クラス
オプション。生成される入力フィールドの表題に関連する CSS スタイル・

クラスです。デフォルト値は **HATSCAPTION** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス
オプション。生成されるテーブルに関連する CSS スタイル・クラスです。入力フィールドの複数の集合がレンダリングされる場合、1 つの HTML テーブルが生成され、それらの入力フィールドを囲みます。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

スタイル
オプション。オーバーライドする任意の CSS プロパティー。例えば、このフィールドに「**font-color: red; font-size: 18pt;**」と指定して、このウィジェットのフォントの色およびサイズを変更できます。入力したプロパティーは、このウィジェットの各エレメントに適用されます。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

チェック・ボックス

チェック・ボックス・ウィジェットは、HTML チェック・ボックスのレンダリングを担当します。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- 入力フィールド
- 項目の選択
- ライト・ペン (選択)
- 選択フィールド (ENPTUI)

次の図に、変換においてヒント付き入力フィールド・コンポーネントの例の入力フィールドのデータを入力として使用するチェック・ボックス・ウィジェットの表示例を示します。

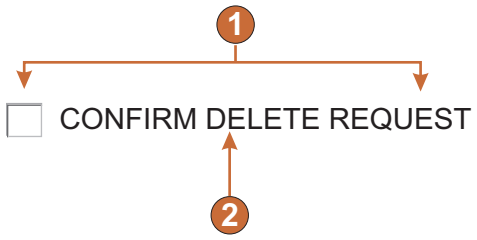


図 34. チェック・ボックス・ウィジェットの例

1. 1 行あたりの列数 (1)
2. 表題 (コンポーネントから)

以下の設定が、このウィジェットに対して構成できます。

選択された値
必須。チェック・ボックスの選択時にホスト画面の入力フィールドに挿入されるストリングです。この値は、Web ページのロード時のチェック・ボックスの初期状態を設定するためにも使用されます。

選択解除された値
必須。チェック・ボックスの選択解除時にホスト画面の入力フィールドに挿入されるストリングです。この値は、Web ページのロード時のチェック・ボックスの初期状態を設定するためにも使用されます。

表題ソース (Caption source)
生成されるチェック・ボックスの表題を決定する方法を指定します。オプション: 「**コンポーネントから**」 (コンポーネントから取り出した表題を使用) または「**カスタム**」(指定された表題を使用)。

カスタム表題
オプション。チェック・ボックスの表題を指定します。

注: チェック・ボックスの表題を生成しない場合は、この設定をブランクのままにしてください。

表題のスペースをトリム (Trim spaces on caption)
これを選択した場合、表題の両端から空白 (余分なスペース) が除去されます。

1 行あたりの列数
次の行に折り返されるまでに、水平方向に表示されるチェック・ボックスの数です。

注:

1. この設定はデフォルト・レンダリングでは適用されません。
2. ウィジェットのプレビューは必ずしも全ページのプレビューと一致するとは限りません。これはウィジェットのプレビューが単にコンポーネントの出力をレンダリングしているだけであるために起こります。ウィジェットのプレビューは、全ページのプレビューで行われているように、レンダリングしたときに画面の配置を保存しようとはしません。

チェック・ボックスのスタイル・クラス Web のみ

オプション。生成されるチェック・ボックスに関連する CSS スタイル・クラスです。HTML 入力タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSCHECKBOX** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

表題のスタイル・クラス Web のみ

オプション。生成されるチェック・ボックスの表題に関連する CSS スタイル・クラスです。デフォルト値は **HATSCAPTION** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス Web のみ

オプション。生成されるテーブルに関連する CSS スタイル・クラスです。チェック・ボックスが複数レンダリングされる場合、1 つの HTML テーブルが生成され、それらのチェック・ボックスを囲みます。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル Web のみ

オプション。オーバーライドする任意の CSS プロパティー。例えば、このフィールドに「**font-color: red; font-size: 18pt;**」と指定して、このウィジェットのフォントの色およびサイズを変更できます。入力したプロパティーは、このウィジェットの各エレメントに適用されます。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド RCP のみ

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

コンボ RCP のみ

コンボ・ウィジェットは、編集可能なドロップダウン組み合わせボックスをレンダリングします。

注:

1. セキュリティー上の理由から、コンボ・ウィジェットを使用してパスワード・フィールドをレンダリングしないでください。コンボ・ウィジェットでレンダリングしたパスワード・フィールドに文字を入力すると、ユーザー入力文字として表示されます。
2. このウィジェットを使用する際の DBCS に関する考慮事項については、 543 ページの『AutoIME 切り替え』を参照してください。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- コマンド行
- 入力フィールド
- ヒント付き入力フィールド
- 項目の選択

次の図に、変換においてヒント付き入力フィールド・コンポーネントの例のデータを入力として使用するコンボ・ウィジェットの表示例を示します。

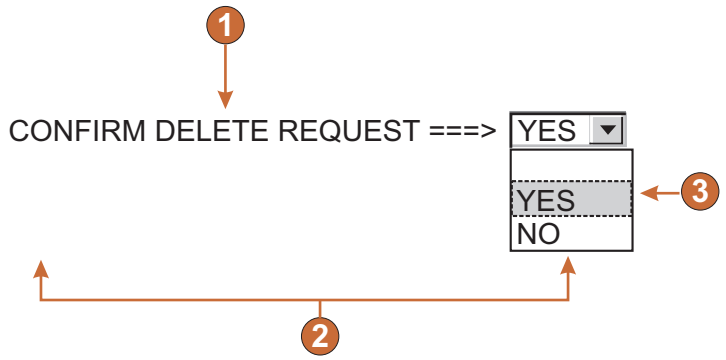


図 35. コンボ・ウィジェットの例

1. 表題 (コンポーネントから)
2. 1 行あたりの列数 (デフォルトは 1)
3. 設定 (ヒントから)

以下の設定が、このウィジェットに対して構成できます。

グローバル変数から設定
これを選択した場合、指定したグローバル変数がドロップダウンに設定されます。

リストの値を含むグローバル変数
値の集合を含む索引付きグローバル変数の名前を指定します。グローバル変数内の各索引に対する項目が、ドロップダウンに作成されます。

共用 HATS のローカル・グローバル変数と共用グローバル変数は同じ名前を持つことができます。共用グローバル変数を使用してリスト項目にデータを取り込む場合は、このボックスを選択します。このボックスをクリアした場合は、ローカル・グローバル変数が使用されます。

リストの表題を含むグローバル変数
オプション。表題の集合を含む索引付きグローバル変数の名前を指定します。この値によって指定されるグローバル変数のサイズは、前の設定で指定されたサイズ以上である必要があります。この索引付きグローバル変数の索引は、グローバル変数の値の索引と一致する必要もあります (その結果、ユーザーに表示される実際の値と表

題が同期します)。この値が指定されない場合は、ドロップダウン中の各項目の表題が、その値になります。

ストリングから設定

これを選択した場合、指定したストリングがドロップダウンに設定されます。

リスト項目

オプション。ドロップダウン・リストに含まれる項目のストリングを指定します。項目はセミコロン (;) で分離する必要があります。リスト項目の表題とリスト項目の値が異なるようにするには、表題と値を等号 (=) で区切って両方を入力します。例えば、**Apple=A;Grape=G** の値は、2 つの項目「**Apple**」と「**Grape**」を持つドロップダウン・リストをレンダリングします。最初の項目を選択すると、関連するホスト画面の入力フィールドに「**A**」が挿入されます。

ドロップダウンの項目とホスト画面に挿入される値を同一にするには、項目のみを入力します。例えば、**Apple=A;G** です。この例では、ドロップダウン・リストとホスト画面の入力フィールドに **G** が表示されます。

ヒントから設定

これを選択した場合、コンポーネントによって認識されたヒントがドロップダウンに設定されます (「ヒント付き入力フィールド」コンポーネントの使用時に限り適用可能)。

表題ソース (Caption source)

生成されるドロップダウンの表題を決定する方法を指定します。オプション: 「**コンポーネントから**」 (コンポーネントから取り出した表題を使用) または「**カスタム**」(指定された表題を使用)。

カスタム表題

オプション。ドロップダウンの表題を指定します。

注: ドロップダウンの表題を生成しない場合は、この設定をブランクのままにしてください。

1 行あたりの列数

次の行に折り返されるまでに、水平方向に表示されるドロップダウンの数です。デフォルトは 1 です。

注:

1. この設定はデフォルト・レンダリングでは適用されません。
2. ウィジェットのプレビューは必ずしも全ページのプレビューと一致するとは限りません。これはウィジェットのプレビューが単にコンポーネントの出力をレンダリングしているだけであるために起こります。ウィジェットのプレビューは、全ページのプレビューで行われているように、レンダリングしたときに画面の配置を保存しようとはしません。

最大長をオーバーライド

これを選択した場合、各生成入力フィールドに指定された最大長の値が設定されます。この設定は、入力フィールドに入力可能な文字数を手動で制限す

る場合に便利です。クリアした場合は、入力文字の最大許容数は、認識する側のコンポーネントによって決定されます。

最大長 各入力フィールドに入力可能な最大文字数です。

前景色を使用可能にする
これを選択した場合、ホスト画面の前景色がレンダリングされます。

色は、リッチ・クライアント・テンプレートによってマップされます。

注: この設定は、「**色のオーバーライド**」>「**前景色**」の設定によってオーバーライドされます。

拡張属性を使用可能にする
これを選択した場合、拡張フィールド属性 (明滅、反転表示、下線、および列分離子) が表示されます。

拡張フィールド属性は、リッチ・クライアント・テンプレートによってマップされます。

フォントのオーバーライド
デフォルトのフォント名、フォント・スタイル、およびフォント・サイズをオーバーライドするには、このオプションを選択し、提供される「**フォント**」 設定コントロールを使用します。

色のオーバーライド
前景色と背景色をオーバーライドするには、このオプションを選択し、提供される「**前景色**」および「**背景色**」設定コントロールを使用します。

ダイアログ

ダイアログ・ウィジェットは、ダイアログ・ボックスをレンダリングします。ダイアログ・コンポーネントの提供するデータをレンダリングします。

次の図に、変換においてダイアログ・コンポーネントの例のデータを入力として使用するダイアログ・ウィジェットの表示例を示します。

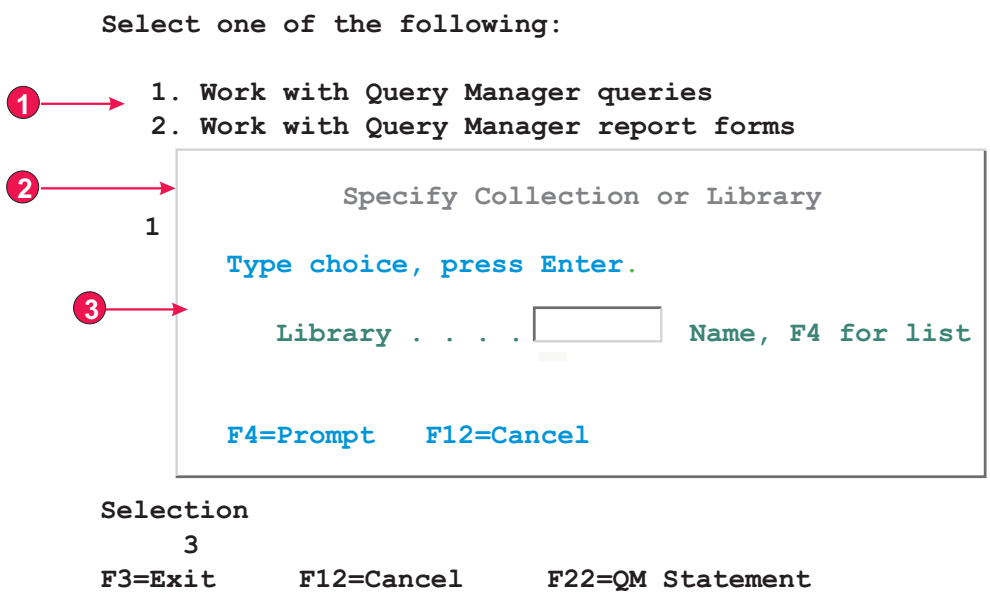


図 36. ダイアログ・ウィジェットの例

1. 使用不可領域
2. 特殊な境界
3. 除去された境界文字

以下の設定が、このウィジェットに対して構成できます。

ダイアログ外部の使用不可領域の非表示
ダイアログの外側の領域を非表示にするかどうかを指定します。

使用不可エリアのスタイル・クラス (Disabled area style class) Web のみ
使用不可エリアで使用する CSS スタイル・クラスを指定します。

特別な境界の使用 Web のみ
元のテキストの代わりに特別な境界を使用するかどうかを指定します。

境界文字の除去 (Remove border characters) Web のみ
元の境界文字を表示するかどうかを指定します。

境界のスタイル・クラス (Border style class)
境界で使用する CSS スタイル・クラスを指定します。

境界のスタイル Web のみ

左方 左の境界で使用する CSS スタイル・クラスを指定します。

右方 右の境界で使用する CSS スタイル・クラスを指定します。

上部 上の境界で使用する CSS スタイル・クラスを指定します。

下部 下の境界で使用する CSS スタイル・クラスを指定します。

コーナーのスタイル Web のみ

左上 境界の左上隅で使用する CSS スタイル・クラスを指定します。

右上 境界の右上隅で使用する CSS スタイル・クラスを指定します。

左下 境界の左下隅で使用する CSS スタイル・クラスを指定します。

右下　境界の右下隅で使用する CSS スタイル・クラスを指定します。

ドロップダウン (データ入力)

ドロップダウン (データ入力) ウィジェットは、ドロップダウン・リストのレンダリングを担当します。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- コマンド行
- 入力フィールド
- ヒント付き入力フィールド
- 項目の選択

次の図に、変換において ヒント付き入力フィールド・コンポーネントの例のデータを入力として使用するデータ入力ドロップダウン・リスト・ウィジェットの表示例を示します。

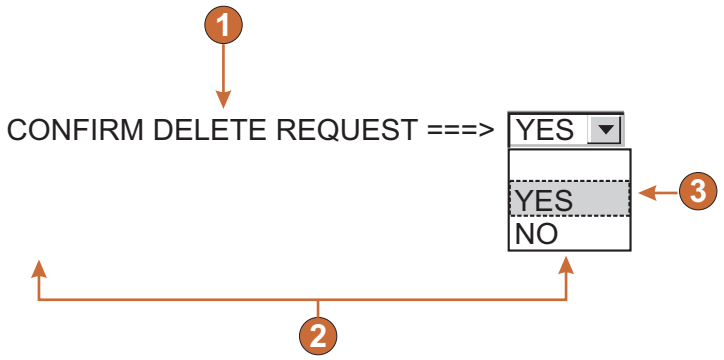


図 37. データ入力ドロップダウン・リスト・ウィジェットの例

1. 表題 (コンポーネントから)
2. 1 行あたりの列数 (1)
3. 設定 (ヒントから)

以下の設定が、このウィジェットに対して構成できます。

グローバル変数から設定
これを選択した場合、指定したグローバル変数がリストに設定されます。

リストの値を含むグローバル変数
値の集合を含む索引付きグローバル変数の名前を指定します。グローバル変数内の各索引の項目がドロップダウン・リストに作成されます。

共用　HATS のローカル・グローバル変数と共用グローバル変数は同じ名前を持つことができます。共用グローバル変数を使用してリスト項

目にデータを取り込む場合は、このボックスを選択します。このボックスをクリアした場合は、ローカル・グローバル変数が使用されます。

リストの表題を含むグローバル変数
オプション。表題の集合を含む索引付きグローバル変数の名前を指定します。この値によって指定されるグローバル変数のサイズは、前の設定で指定されたサイズ以上である必要があります。この索引付きグローバル変数の索引は、グローバル変数の値の索引と一致する必要もあります (その結果、ユーザーに表示される実際の値と表題が同期します)。この値が指定されない場合は、ドロップダウン・リスト内の各項目の表題がその値になります。

ストリングから設定
これを選択した場合、指定したストリングがリストに設定されます。

リスト項目
オプション。ドロップダウン・リストに含まれる項目のストリングを指定します。項目はセミコロン (;) で分離する必要があります。リスト項目の表題とリスト項目の値が異なるようにするには、表題と値を等号 (=) で区切って両方を入力します。例えば、**Apple=A;Grape=G** の値は、2 つの項目「**Apple**」と「**Grape**」を持つドロップダウン・リストをレンダリングします。最初の項目を選択すると、関連するホスト画面の入力フィールドに「**A**」が挿入されます。

ドロップダウン・リストの項目とホスト画面に挿入される値を同一にするには、項目のみを入力します。例えば、**Apple=A;G** です。この例では、ドロップダウン・リストとホスト画面の入力フィールドに **G** が表示されます。

ヒントから設定
これを選択した場合、コンポーネントに認識されたヒントがドロップダウン・リストに設定されます (「ヒント付き入力フィールド」コンポーネントの使用時に限り適用可能)。

表題ソース (Caption source)
生成されるドロップダウン・リストの表題を決定する方法を指定します。オプション: 「**コンポーネントから**」 (コンポーネントから取り出した表題を使用) または「**カスタム**」(指定された表題を使用)。

カスタム表題
オプション。ドロップダウン・リストの表題を指定します。

注: ドロップダウン・リストの表題を生成しない場合は、この設定はブランクのままにしてください。

表示可能な行数
ユーザーに対してドロップダウン・リスト内に表示される行の行数。リスト内の項目数がこの数よりも多い場合は、スクロール・バーが表示されます。これにより、ユーザーはリスト全体をスクロールできます。デフォルトは **1** です。

1 行あたりの列数

次の行に折り返されるまでに、水平方向に表示されるドロップダウン・リストの数です。

注:

1. この設定はデフォルト・レンダリングでは適用されません。
2. ウィジェットのプレビューは必ずしも全ページのプレビューと一致するとは限りません。これはウィジェットのプレビューが単にコンポーネントの出力をレンダリングしているだけであるために起こります。ウィジェットのプレビューは、全ページのプレビューで行われているように、レンダリングしたときに画面の配置を保存しようとはしません。

選択時に自動的に実行依頼

これを選択した場合、ドロップダウン・リストで選択操作が行われると、その選択内容が実行依頼されます。

送信ボタンを表示

これを選択した場合、ドロップダウン・リストの横に送信ボタンがレンダリングされます。

送信ボタンの表題

オプション。送信ボタンの表題を指定します。

ドロップダウンのスタイル・クラス Web のみ

オプション。生成されたドロップダウン・リストに関連する CSS スタイル・クラスです。HTML ドロップダウン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSDROPDOWN** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

リスト・オプションのスタイル・クラス Web のみ

オプション。ドロップダウン・リストの各オプションに関連する CSS スタイル・クラスです。デフォルト値は **HATSOPTION** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

表題のスタイル・クラス Web のみ

オプション。生成されるドロップダウン・リストの表題に関連する CSS スタイル・クラスです。デフォルト値は **HATSCAPTION** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス Web のみ

オプション。生成されるテーブルに関連する CSS スタイル・クラスです。複数のドロップダウン・リストがレンダリングされる場合、HTML テーブルはこれらのドロップダウン・リストを囲むように生成されます。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

スタイル Web のみ

オプション。オーバーライドする任意の CSS プロパティー。例えば、このフィールドに「**font-color: red; font-size: 18pt;**」と指定して、このウィジェットのフォントの色およびサイズを変更できます。入力したプロパティーは、このウィジェットの各エレメントに適用されます。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・

ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド RCP のみ
デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド RCP のみ
前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

ドロップダウン (選択)

ドロップダウン (選択) ウィジェットは、HTML ドロップダウン・リストのレンダリングを担当します。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- ファンクション・キー
- 選択リスト

次の図に、変換において選択リスト・コンポーネントの例のデータを入力として使用する選択リスト・ドロップダウン・リスト・ウィジェットの表示例を示します。

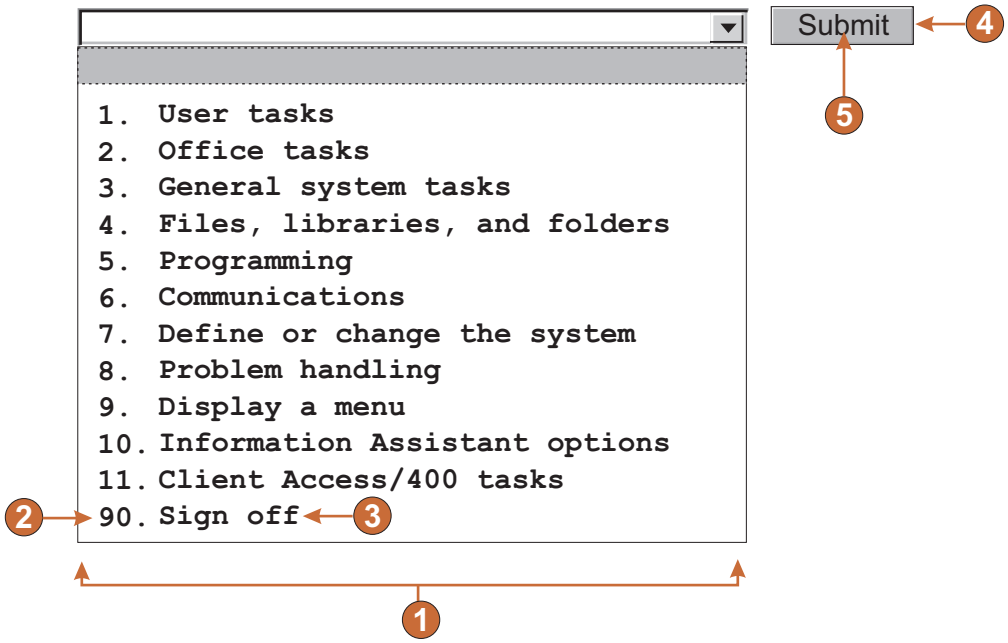


図 38. 選択リスト・ドロップダウン・リスト・ウィジェットの例

1. 表題のタイプ (両方を表示)

2. 先行トークン (両方を表示)
3. 説明 (両方を表示)
4. 送信ボタン (送信ボタンを表示)
5. 送信ボタンの表題 (送信)

以下の設定が、このウィジェットに対して構成できます。

表題のタイプ
リスト項目の先行トークンの表示方法を指定します。コンポーネントから先行トークンの値および説明が取り込まれます。つまり、ドロップダウン・リスト内に表示される項目を選択できます。例えば、ホスト画面に「**4. メール**」というメニュー項目が存在する場合、表題表示として「**4**」、「**メール**」、または「**4. メール**」を選択できます。

選択時に自動的に実行依頼
これを選択した場合、ドロップダウン・リストで選択操作が行われると、その選択内容が実行依頼されます。ドロップダウン・リストに、ファンクション・キーが含まれている場合は、選択したリストの項目のキーがホストに送信されます。ドロップダウン・リストに選択リスト項目が含まれている場合は、選択した項目の値がターゲット・フィールドに挿入され、指定したアクション・キーがホストに送信されます。

送信ボタンを表示
これを選択した場合、ドロップダウン・リストの横に送信ボタンがレンダリングされます。ファンクション・キー・コンポーネントとこのウィジェットを併用しているときには、適用しないでください。

送信ボタンの表題
オプション。送信ボタンの表題を指定します。

ボタンのスタイル・クラス **Web のみ**
オプション。生成される送信ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ドロップダウンのスタイル・クラス **Web のみ**
オプション。生成されたドロップダウン・リストに関連する CSS スタイル・クラスです。HTML ドロップダウン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSDROPDOWN** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

リスト・オプションのスタイル・クラス **Web のみ**
オプション。ドロップダウン・リストの各オプションに関連する CSS スタイル・クラスです。デフォルト値は **HATSOPTION** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

表題のスタイル・クラス **Web のみ**
オプション。生成されるドロップダウン・リストの表題に関連する CSS スタイル・クラスです。デフォルト値は **HATSCAPTION** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス Web のみ

オプション。生成されるテーブルに関連する CSS スタイル・クラスです。複数のドロップダウン・リストがレンダリングされる場合、HTML テーブルはこれらのドロップダウン・リストを囲むように生成されます。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

スタイル Web のみ

オプション。オーバーライドする任意の CSS プロパティー。例えば、このフィールドに「**font-color: red; font-size: 18pt;**」と指定して、このウィジェットのフォントの色およびサイズを変更できます。入力したプロパティーは、このウィジェットの各エレメントに適用されます。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド RCP のみ

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド RCP のみ

前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

フィールド

フィールド・ウィジェットは、フィールド・コンポーネントによって認識される入力フィールドとテキストのレンダリングを担当します。

注: このウィジェットを使用する際の DBCS に関する考慮事項については、548 ページの『SBCS での最大長の除去』を参照してください。

次の図に、変換においてフィールド・コンポーネントの例のデータを入力として使用するフィールド・ウィジェットの表示例を示します。

図 39. フィールド・ウィジェットの例

1. 保護テキスト

フィールド・ウィジェットのテーブル・レイアウトの主要な目標は、GUI でホスト画面をレンダリングするときに、元のホスト画面のレイアウトを維持することで

す。デフォルト・レンダリングでは、テーブルを使用して画面のレイアウトが維持されます。このテーブルは通常、80 列および 24 行で構成されます。フィールド・ウィジェットの各種位置合わせ設定 (「**通常**」、「**文字ごと Web のみ**」、および「**ワードごと**」) により、このテーブルのテーブル・セルの作成方法が変更されます。

通常の位置合わせでは、各ホスト・フィールドはテーブル内のテーブル・セルにより表わされます。セルが 1 つ以上の列を使用することがあります。詳細については、「colspan」HTML 属性を参照してください。ホスト画面上で列 10 から開始するフィールドは、テーブル内の列 10 に位置付けられます。通常の位置合わせでは、各フィールドが正しい列位置から開始されることが保証されます。

グローバル規則では一般に、変換元のフィールドよりも多くのページ・スペースが使用されるため、グローバル規則がページの位置合わせに影響することがあります。例えば、長さ 5 文字の入力フィールドがあるとします。デフォルト・レンダリングでは、このフィールドに対しテーブル内で 5 列が割り振られます。6 文字以上のスペースを使用するグローバル規則によりこのフィールドがレンダリングされると、このフィールドが位置するテーブル・セルが強制的に拡張されます。つまり、セルの幅が増加します。列全体が影響を受けるため、これはテーブル全体に影響します。一般に、どの行においても、フィールドの右側のセルは右側に移動します。これは、変換されたフィールドが含まれている列の幅が大きくなったためです。したがって、行 5 の列 60 から始まるフィールドは常に、行 20 の列 60 から開始されるフィールドに位置合わせされます。

文字ごとの位置合わせは、HATS Web プロジェクトでのみサポートされています。この位置合わせオプションを選択すると、入力フィールド以外のフィールドの各文字が、それぞれ個別のテーブル・セルの中でレンダリングされます。このオプションを使用すると、画面上のすべての文字が正しい開始位置に合わせられますが、生成されるページのサイズが非常に大きくなります。

ワードごとの位置合わせでは、画面上の各ワードがそれぞれ個別のテーブル・セルの中でレンダリングされます。つまり、複数のワードからなる単一フィールドは、複数のテーブル・セルでレンダリングされます。これにより、画面上のすべてのワードが正しい位置でレンダリングされます。このオプションは一般に、ホスト画面全体でさまざまな入力フィールドが分散している場合に使用します。入力フィールドでは通常のテキストよりも多くのページ・スペースが使用されるため、入力フィールドが原因でページが不適切に位置合わせされることがあります。

次の例に、デフォルト・レンダリングでの各種位置合わせオプションによるフィールドおよび文字のテーブル・セルへのマッピングを示します。実際のデフォルト・レンダリングでは、テーブル・セルは不可視です。

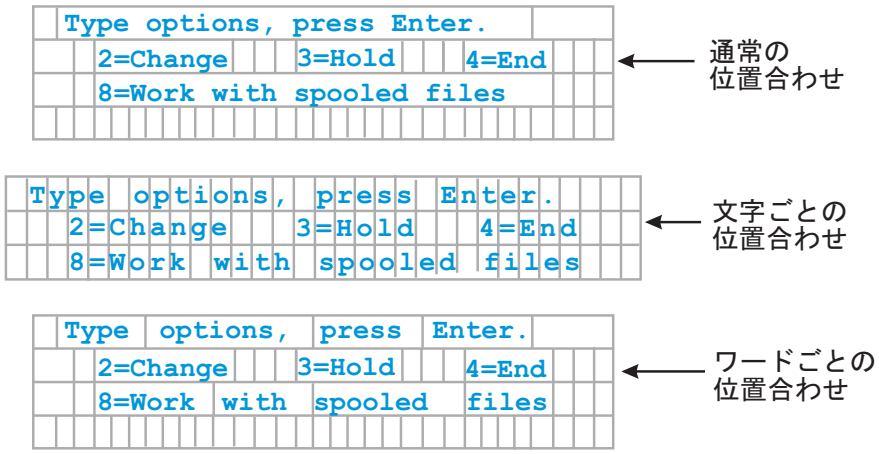


図 40. フィールド・ウィジェットの位置合わせの例

以下の設定が、このウィジェットに対して構成できます。

読み取り専用

これを選択した場合、入力フィールドはレンダリングされません。入力フィールドにテキストが含まれている (かつパスワード保護されていない) 場合、代りにそのテキストが表示されます。この設定は、ページの情報を表示するが編集させたくない場合に便利です。

レイアウト

この設定を使用して、このウィジェットによりレンダリングされる出力のレイアウトを指定します。元のホスト画面のレイアウトを保存するために、「**テーブル**」を選択して、テーブルの出力をレンダリングします。 これは、モバイル装置に対して最適化されていない Web アプリケーションのデフォルトです。HTML の量とブランク・スペースを削減するために、「**分離**」を選択し、インライン・スパン・タグを使用して出力をレンダリングし、フィールドを区別します。これは、モバイル装置に対して最適化されている Web アプリケーションのデフォルトです。

位置合わせ

テーブル・レイアウトを使用する場合、位置合わせを改善するため、HATS でのテキスト文字をテーブル・セルへマップする方法 (「**通常**」、「**文字ごと**」**Web のみ**、または「**ワードごと**」) を指定します。

テーブルのスタイル・クラス **Web のみ**

オプション。生成されるテーブルに関連する CSS スタイル・クラスです。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSFIELDTABLE** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

nowrap **Web のみ**

位置合わせ設定として「**ワードごと**」を指定すると、変換時に複数フィールドがワード単位で位置合わせされます。ただし、別のウィジェットによる出力が原因で、デフォルト・レンダリング・テーブ

ルのセルが圧縮され、テキストが折り返すことがあります。テキストが折り返さないようにするため、ワードごとの位置合わせを使用する場合は nowrap フラグが HTML 出力に自動的に追加されます。HTML 出力への nowrap フラグの自動追加を無効にするには、この **nowrap** 設定を使用します。

位置合わせ設定をどの値に設定する場合でも、**nowrap** を **false** に設定すると、フィールド・ウィジェットの HTML 出力に nowrap フラグが追加されず、セル内のテキストが折り返し可能になります。ワードごとの位置合わせを使用し、継続的に位置合わせの問題が発生する場合にのみ、**nowrap** を **false** に設定します。

デフォルト・レンダリングでは、テーブル・セルが折り返すと、垂直位置合わせおよび水平位置合わせの両方に悪影響が及び、不適切な出力が作成される原因となります。

位置合わせ設定をどの値に設定する場合でも、**nowrap** を **true** に設定すると、テーブル・セル内でテキストが次の行に折り返すことがありません。

注: GUI では **nowrap** 設定は構成可能な設定として示されません。これは application.hap ファイルのソースで設定する必要があります。例えば、**nowrap** を **false** に設定するには、「HATS Projects」ビューのプロジェクト・フォルダーの「**プロジェクト設定**」ファイルをダブルクリックし、application.hap ファイルのエディターを起動します。「**ソース**」タブで、<classSettings> タグの中のクラス name="com.ibm.hats.transform.widgets.FieldWidget" を見つけ、次の設定を追加します。

```
<setting name="nowrap" value="false"/>
```

保護フィールド上でのカーソルの位置決めを許可します

これを選択した場合、保護フィールドの 1 つをクリックすると、ホスト・カーソルがその特定の位置に設定されます。この設定は、ホスト・アプリケーションがカーソル位置に依存し、ホスト・アプリケーションでカーソルを保護フィールド上に配置する必要がある場合に便利です。

Web アプリケーションの場合は、カーソル位置設定の精度は、「**位置合わせ**」設定（「**通常**」、「**文字ごと Web のみ**」、または「**ワードごと**」）によって決まります。

リッチ・クライアント・アプリケーションの場合は、フィールド内の任意の位置にカーソル位置を設定できます。「**位置合わせ**」設定は、カーソル位置の設定の精度に影響しません。

ブラウザーでの実行時に、これらのフィールドの 1 つがクリックされると、HATS に対してその旨が通知されます。HATS はカーソル位置を内部で更新し、ユーザーに対して新しいカーソル位置を視覚的に示します。デフォルトでは、テキストの背景色を黄色に設定することで視覚的に示します。Web アプリケーションの場合は、開発者が CSS スタイル **HCURSORINDICATOR** を使用して視覚的な表示を制御できます。詳細については、374 ページの『スタイル・シートの使用』を参照してください。この設定を有効にすると、以下の操作が可能になります。

- 保護フィールドの領域をクリックしてカーソルの位置を決めることができる。
- カーソルが無保護フィールド内にない場合でも、ページの読み込み時にカーソルの初期位置を確認できる。

注: フィールド・ウィジェットを使用して画面上の同一領域が複数回レンダリングされる場合は、この設定のみでは正しく機能しません。例えば、タブ付きフォルダーの複数のタブで画面上の同一領域がレンダリングされる場合は、さらに、ユーザーが画面上のその領域にカーソルの位置を設定できるようにするため、保護フィールドをリンクとしてレンダリングする必要があります (以下の「**リンクを使用するタブ操作を許可**」を参照)。

リンクを使用するタブ操作を許可 Web のみ

これを選択した場合、各保護フィールドは HTML リンクとしてレンダリングされます。これにより、保護フィールド間でのタブ操作が可能になります。カーソル位置は下線付き文字として表示されます。

注: このオプションを選択すると、ページの各保護フィールドに対して HTML アンカー <A> タグが生成されます。このため、Web ページのサイズが大きくなります。

生成される Web ページのサイズが不必要に大きくなることを防ぐため、デフォルトではこのオプションは選択されていません。

リンクのスタイル・クラス Web のみ

生成される各リンクに関連する CSS スタイル・クラスです。HTML リンク・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSPROTLINK** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

アクション・キー

オプション。保護フィールドの選択時、ホストに送信されるホスト AID キー・ニーモニックを指定します。例えば、**[enter]** はカーソルを指定の位置に設定し、enter を送信します。また、**[pf1]** は、カーソルを指定の位置に設定し、pf1 を送信します。

リンクのスペースをトリム (Trim spaces on links)

これを選択した場合、保護フィールド・リンクの両端から空白 (余分なスペース) が除去されます。この設定は、リンクとしてレンダリングされるテキストの容量を減少する場合に便利です。

入力フィールドのカーソル位置オプションを使用可能にする Web のみ

このボックスを選択して、ユーザーが入力フィールドでデータ入力モードからカーソル位置決めモードに切り替えができるようにします。データ入力モードのときは、ユーザーは入力フィールドにデータを入力できます。カーソル位置決めモードのときは、タブ移動または別の方法で、入力フィールドの任意の文字上にカーソルを配置できます。この設定は他のカーソル位置決め機能を持たないモバイル装置用に最適化されている Web プロジェクトでは

便利です。HATS Toolkit GUI に表示されないこのオプションの他の設定のついては、 54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

前景色を使用可能にする

これを選択した場合、ホスト画面の前景色がレンダリングされます。

Web アプリケーションの場合は、前景色はその色を表す CSS スタイルシート・クラスにマップされます。例えば、ホスト画面フィールドが **RED** とマークされている場合、フィールド・ウィジェットは、そのフィールド用に生成された HTML を、クラス名属性が **HRED** に設定されているタグで囲みます。この処理によって、ユーザーは、生成される Web ページ上にホスト画面の色を再マップすることができます。

注: blacktheme.css は、ホスト画面フィールドの前景色を直接マップします (例えば、ホスト画面上の赤色のテキストは、生成された Web ページ上で赤色のテキストとして表示されます)。ただし、monochrome.css などの他のスタイルシート・ファイルは、フィールドの色を別の方法でマップし、一貫性のある、最新のスタイルを作成しようとします。

リッチ・クライアント・アプリケーションの場合、色はリッチ・クライアント・テンプレートによってマップされます。

注: この設定は、「**色のオーバーライド**」>「**前景色**」の設定によってオーバーライドされます。

フィールドのスタイル・クラス **Web のみ**

オプション。生成されたフィールドに関連する CSS スタイル・クラスです。HTML タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSFIELD** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

拡張属性を使用可能にする

これを選択した場合、拡張フィールド属性 (明滅 Web のみ、反転表示、下線、および列分離子) がレンダリングされます。また、3270 Web アプリケーションの場合、拡張フィールドの色もマップされます (詳細については、「**前景色を使用可能にする**」の設定を参照してください)。

リッチ・クライアント・アプリケーションの場合は、反転表示のレンダリングでは、背景色としてリッチ・クライアント・テンプレートでマップされている色が使用され、前景色として黒が使用されます。列分離子は、フィールドの下の破線としてレンダリングされます。

明滅スタイル **Web のみ**

ホスト画面における明滅テキストのレンダリングに使用する CSS スタイル設定。

反転表示スタイル (Reverse video style) **Web のみ**

ホスト画面における反転表示テキストのレンダリングに使用する CSS スタイル設定。

下線スタイル (Underline style) **Web のみ**

ホスト画面における下線付きテキストのレンダリングに使用する CSS スタイル設定。

列分離子スタイル (Column separator style) **Web のみ**
ホスト画面における列分離子のレンダリングに使用する CSS スタイル設定。

詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

モノスペース・フォントを使用してレンダリング
選択されている場合、フィールドはモノスペース・フォントを使用してレンダリングされます。選択されていない場合、ロケール環境のデフォルト・フォントを使用してレンダリングされます。

3270 数値ロックを有効にする
選択されている場合、ユーザーは、0 から 9 までの数字、正符号、負符号、ピリオド、コンマ、およびヒンディ数字のみを、3270 数値フィールドに入力できます。クリアされている場合、任意の文字を入力できます。デフォルトではクリアされています。

注: Web アプリケーションの場合、必要に応じて lxgwfunctions.js ファイルの allowNumLockOnly() 関数および allowNumLockOnlyForIEMobile() 関数で文字コード値を使用して、受け入れられるキーのリストをカスタマイズできます。

入力フィールドの下線文字を除去 (Strip underscores on input field)
テキストをレンダリングするときにテキストから下線文字を除去する場合は、このボックスを選択します。

入力フィールドのスペースをトリム
入力フィールドから先行および後続のスペースをトリムする場合は、このボックスを選択します。

スタイル **Web のみ**
オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

注: このウィジェットによって多数のタグが生成されるため、この設定に値を指定すると、ページのダウンロード時間が大幅に増加する場合があります。囲んでいる変換または組み込みスタイルシート・ファイルにスタイル・オーバーライドを指定します。

フォントのオーバーライド **RCP のみ**
デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド **RCP のみ**
前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

グラフ (水平棒、折れ線、垂直棒)

グラフ・ウィジェットは、JPEG グラフ・イメージのレンダリングを担当します。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- テーブル
- テーブル (フィールド)
- テーブル (可視)

注:

1. プロキシー・サーバーを介して HATS アプリケーションがアクセスされたときにグラフ・ウィジェットを表示するには、プロキシー・サーバーを使用するよう HATS アプリケーションを構成する必要があります。手順については、44 ページの『プロキシー・サーバーを使用するための HATS アプリケーションの構成』を参照してください。
2. このウィジェットは、非常に大きな範囲のデータをレンダリングするには、適していない場合があります。

変換においてテーブル (可視)・コンポーネントの例のデータを入力として使用する水平棒グラフ、折れ線グラフ、および垂直棒グラフの表示例を次に示します。

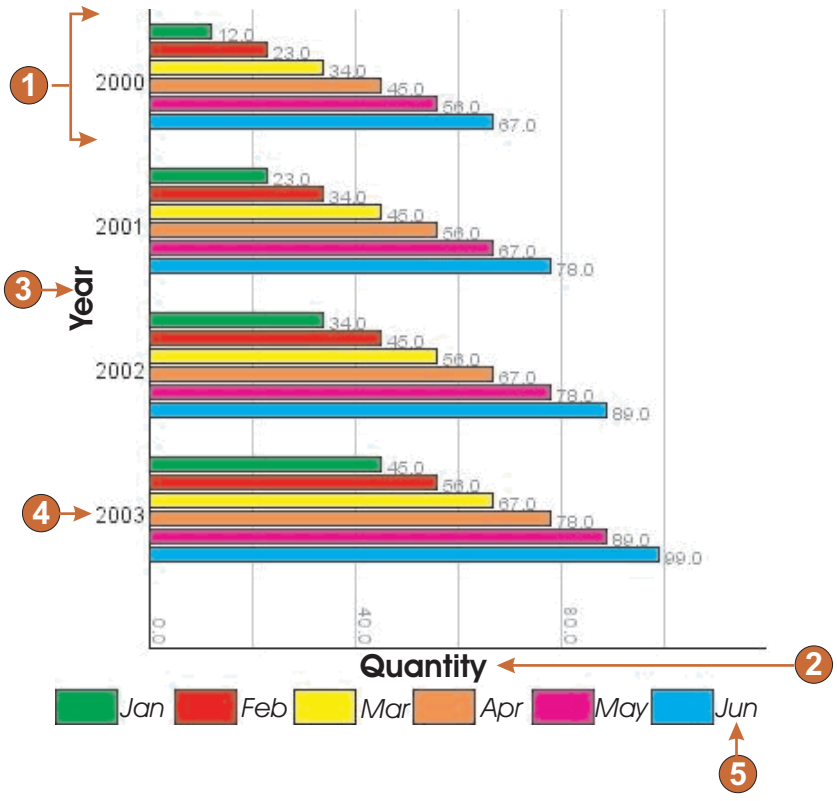


図 41. 水平棒グラフ・ウィジェットの例

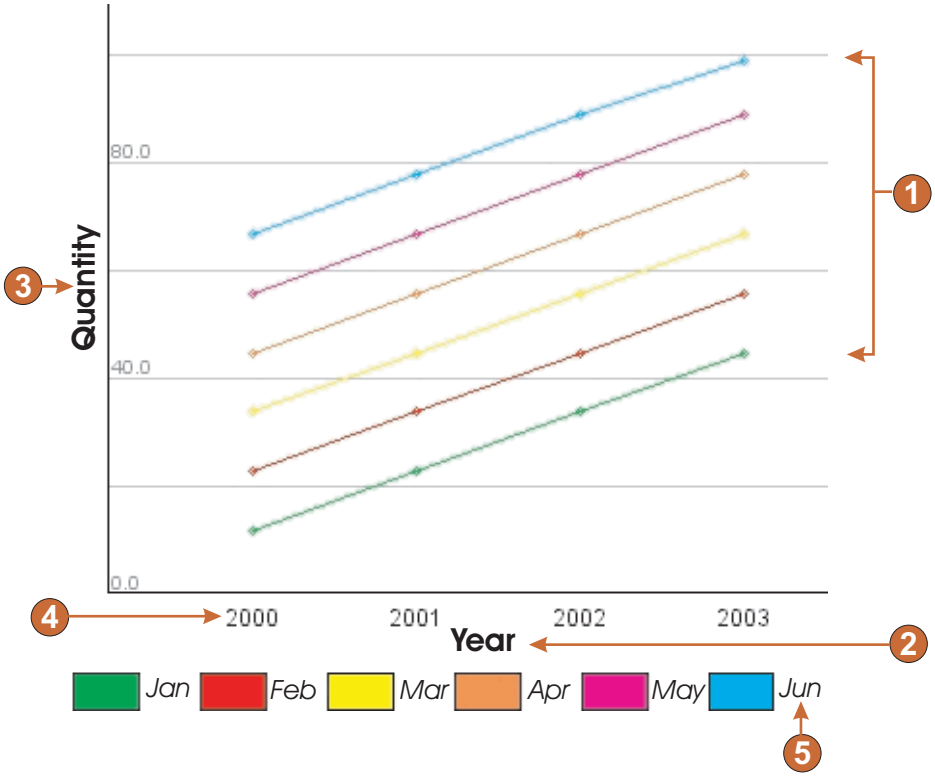


図 42. 折れ線グラフ・ウィジェットの例

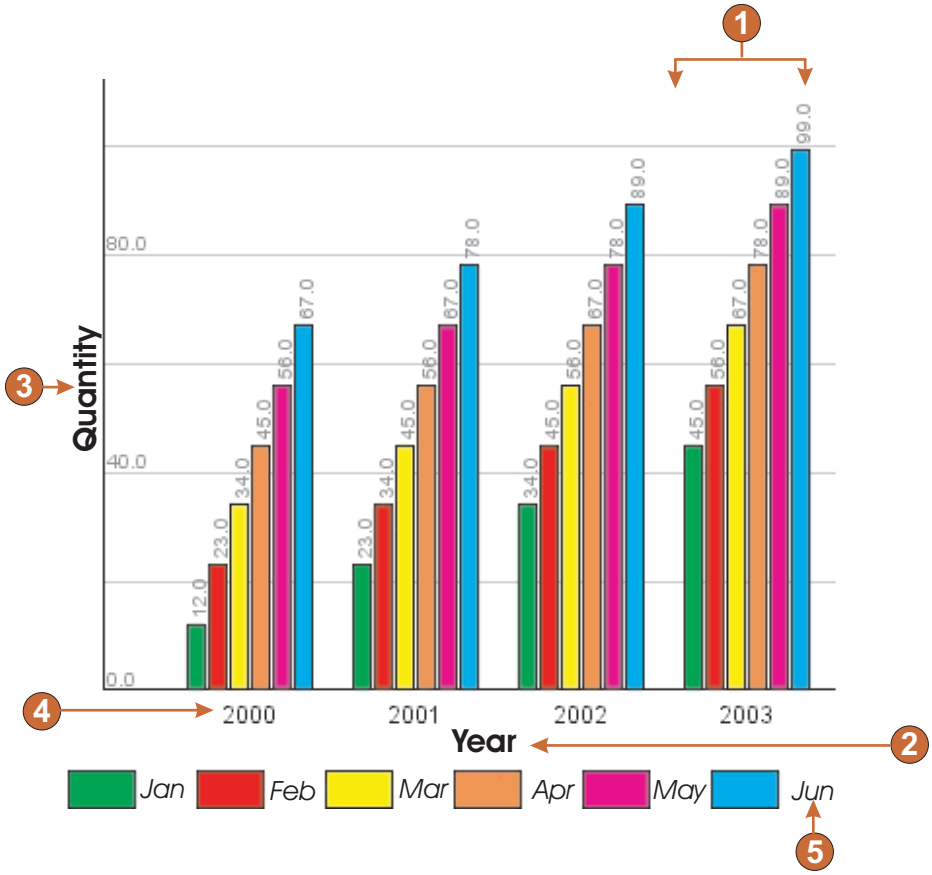


図 43. 垂直棒グラフ・ウィジェットの例

1. データ・セット (6)
2. X 軸のタイトル (年、水平棒グラフ上では数量)
3. Y 軸のタイトル (数量、水平棒グラフ上では年)
4. データ・ポイント・ラベル (抽出)
5. データ・セット・ラベル (抽出)

以下の設定が、このウィジェットに対して構成できます。

データ・セットの数
グラフに描画するデータ・セットの数を指定します。この設定は、「**ホスト・コンポーネントを挿入**」ウィザードでのみ表示されます。

データ・セットのソース
1 つ以上のデータ・セットに対する値を取得する方法を指定します。オプション: 「**行**」(テーブルの各行がグラフ化する 1 つのデータ・セットで構成される) または「**列**」(テーブルの各列がグラフ化する 1 つのデータ・セットで構成される)。

幅 (ピクセル)
グラフのピクセル単位の幅。

高さ (ピクセル)
グラフのピクセル単位の高さ。

背景色 グラフの背景に使用する色を指定します。色セレクター・ボタンをクリックして、カラー・パレットを表示します。

背景イメージ
グラフの背景に使用するイメージの名前 (Web プロジェクトの Web Content¥Common¥Images フォルダーまたはリッチ・クライアント・プロジェクトの Rich Client Content¥Images フォルダーからの相対パス) を入力します。「**参照**」ボタンをクリックして、使用中のシステムからイメージをインポートします。

デフォルト・フォント (Default Font)
フォント、タイプ、およびフォント・サイズを指定します。

X 軸のタイトル
オプション。グラフの X 軸のラベルを指定するストリングです。

Y 軸のタイトル
オプション。グラフの Y 軸のラベルを指定するストリングです。

軸の色 グラフの軸を描画するときの色を指定します。色セレクター・ボタンをクリックして、カラー・パレットを表示します。

ラベルの色 (Label color)
グラフのラベルを描画するときの色を指定します。色セレクター・ボタンをクリックして、カラー・パレットを表示します。

テキストのエイリアス除去
これを選択した場合、テキストのエイリアス除去が使用され、描画されるテキストの端が平滑化されます。

代替テキスト
オプション。ユーザーのクライアント・ブラウザーがレンダリングされたグラフ・イメージをロードできない場合に表示するテキストを指定します。

「データ・セットの数」設定の場合のように、以下の設定は、「**ホスト・コンポーネントを挿入**」**ウィザード**に限り表示されます。

データ・ポイント・ラベルを抽出
グラフのデータ・ポイントのラベルを、テーブルの行 (または列) から抽出する場合は、このボックスを選択します。

行 (または列) (Row (or Column))
データ・ポイント・ラベルの抽出元である行 (または列) の数を指定します。この入力フィールドのラベル (行または列) は、「**データ・セットのソース**」の設定で指定された値に依存しています。

凡例用にデータ・セットのラベルを抽出
グラフのデータ・セットのラベルを、テーブルの行 (または列) から抽出する場合は、このボックスを選択します。

行 (または列) (Row (or Column))
データ・セット・ラベルの抽出元である行 (または列) の数を指定します。この入力フィールドのラベル (行または列) は、「**データ・セットのソース**」の設定で指定された値に依存しています。

データ・セット
このボタンをクリックして、データ・ソース設定ダイアログを表示します。このダイアログでは、データ・ソースに関する以下の追加設定を指定できます。

データ・セット「n」、「行」、または「列」
これらのフィールドの番号 (「n」) は、データ・セット・ソースの設定で指定された値と一致しています。データ・セットに使用するデータの任意の行または列の番号を入力します。こうすると、グラフ内の一連のデータを再配列または複写することができます。この入力フィールドのラベルの最終部分は、「データ・セットのソース」の設定で指定された値に依存しています。入力フィールド・ラベルは、「データ・セットのソース」の設定で指定された値と一致しています。

色　データ・セット「n」、「行」、「列」のそれぞれの設定にカラー・ボタンがあります。ボタンは、グラフ内のデータ・セットに使用する色を表示しています。データ・セットの色を変更する場合は、ボタンをクリックしてカラー・パレットを表示します。

凡例のラベル
凡例に使用するラベルを指定してください。「**凡例用にデータ・セットのラベルを抽出**」が選択されていない場合のみ使用可能になります。

ラベル

ラベル・ウィジェットは、テキストのレンダリングを担当します。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- テキスト
- HTML DDS キーワード Web のみ

次の図に、変換においてテキスト・コンポーネントの例のデータを入力として使用するラベル・ウィジェットの表示例を示します。

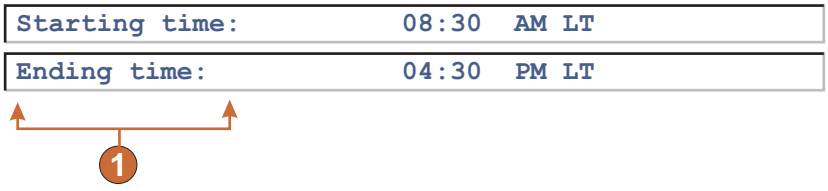


図 44. ラベル・ウィジェットの例

1. ラベル

以下の設定が、このウィジェットに対して構成できます。

スペースのトリム
これを選択した場合、選択されたテキストの両端から空白 (余分なスペース) が削除されます。

スタイル・クラス Web のみ

オプション。生成されるテキストに関連する CSS スタイル・クラスです。デフォルト値は **HATSCAPTION** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル Web のみ

オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド RCP のみ

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド RCP のみ

前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

リンク

リンク・ウィジェットは、HTML リンクのレンダリングを担当します。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- ファンクション・キー
- ライト・ペン (アテンション)
- 選択リスト
- URL
- 選択フィールド (ENPTUI)

次の図に、変換において選択リスト・コンポーネントの例のデータを入力として使用するリンク・ウィジェットの例を示します。

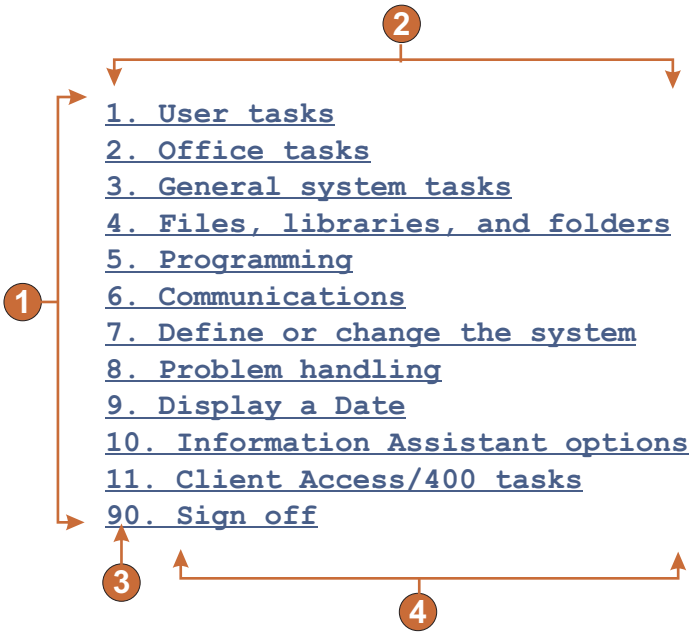


図 45. リンク・ウィジェットの例

1. 行
2. 1 行あたりの列数 (1)
3. 先行トークン (両方を表示)
4. 説明 (両方を表示)

以下の設定が、このウィジェットに対して構成できます。

表題のタイプ
各リンクの表題を決定する方法を指定します。コンポーネントから先行トークンの値および説明が取り込まれます。つまり、リンクのテキストとして表示される項目を選択できます。例えば、ホスト画面に「**4. メール**」というメニュー項目が存在する場合、テキスト表示として「**4**」、「**メール**」、または「**4. メール**」を選択できます。

表題のスペースをトリム (Trim spaces on caption)
これを選択した場合、表題の両端から空白 (余分なスペース) が除去されます。

レイアウト
HTML ページでのリンクの配置方法を指定します。オプション: 「**テーブル**」、「**分離**」。

注:

1. この設定はファンクション・キー・コンポーネントが使用されている場合、デフォルト・レンダリングでは適用されません。
2. ウィジェットのプレビューは必ずしも全ページのプレビューと一致するとは限りません。これはウィジェットのプレビューが単にコンポーネントの出力をレンダリングしているだけであるために起こります。ウィジェットのプレビューは、全ページのプレビューで行われているように、レンダリングしたときに画面の配置を保存しようとはしません。

1 行あたりの列数

次の行に折り返されるまでに、水平方向に表示されるリンクの数です。

分離文字

HTML ページ上でレンダリングされたリンクを区切るために使用する分離文字を指定します。ドロップダウン・リストから選択するか入力フィールドに手動で入力するかを選択できます。

ターゲット (Target) **Web のみ**

リンクされた URL のコンテンツを表示するターゲット (同じ、新規、親、またはトップのウィンドウ) をドロップダウン・リストから選択します。

リンクのスタイル・クラス **Web のみ**

オプション。生成される各リンクに関連する CSS スタイル・クラスです。HTML リンク・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSLINK** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**

オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド **RCP のみ**

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド **RCP のみ**

前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

リンク (項目選択)

リンク (項目選択) ウィジェットは、項目選択コンポーネントをリンクとしてレンダリングします。

次の図に、変換において項目の選択コンポーネントの例のデータを入力として使用するリンク (項目選択) ウィジェットの例を示します。

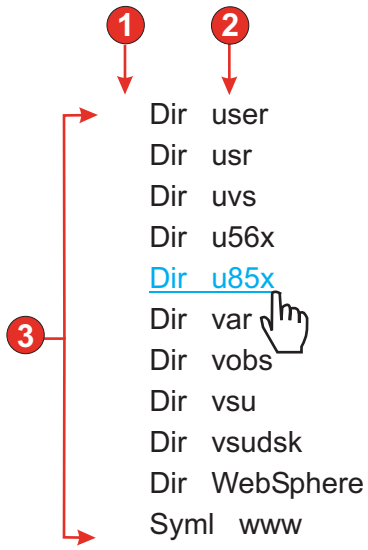


図 46. リンク (項目選択) ウィジェットの例

1. 入力フィールドの表示 (いいえ)
2. 1 行あたりの列数 (デフォルトは 1)
3. レイアウト (テーブル)

以下の設定が、このウィジェットに対して構成できます。

レイアウト
HTML ページでのリンクの配置方法を指定します。オプション: 「**テーブル**」、「**分離**」。

1 行あたりの列数
レンダリングした HTML ページの各行に表示するこのウィジェットのインスタンスの数を指定します。

分離文字
HTML ページ上でレンダリングされたリンクを区切るために使用する分離文字を指定します。ドロップダウン・リストから選択するか入力フィールドに手動で入力するかを選択できます。

入力フィールドの表示
各リンクの入力フィールドを表示するかどうかおよび入力フィールドの位置 (リンクの左側/右側) を指定します。オプション: 「**いいえ**」、「**表題の左**」、「**表題の右**」

選択値 (Selection value)
入力フィールドの左側に配置する文字を指定します。

自動的に送信するキー
ホストに送信する AID キーを指定するか、送信するキーがないことを指定します。

リンクのスタイル・クラス
オプション。生成されるランチャー・リンクに関連する CSS スタイル・クラスです。HTML リンク・タグの class 属性の値には、この値がセットさ

れます。デフォルト値は、**HATSPROTLINK HBLANK** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル
オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

リスト

リスト・ウィジェットは、リスト・ボックスをレンダリングします。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- コマンド行
- 入力フィールド
- ヒント付き入力フィールド
- 項目の選択

次の図に、変換におけるリスト・ウィジェットの表示例を示します。

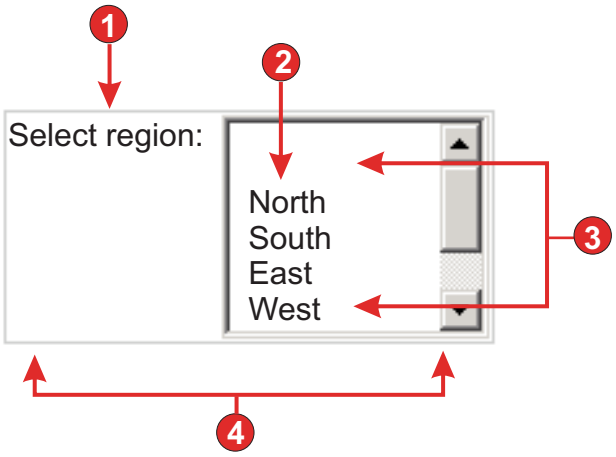


図 47. リスト・ウィジェットの例

1. 表題 (カスタム)
2. 設定 (ストリングから)
3. 表示可能な行数 (デフォルトは 5)
4. 1 行あたりの列数 (デフォルトは 1)

以下の設定が、このウィジェットに対して構成できます。

グローバル変数から設定
これを選択した場合、指定したグローバル変数がリストに設定されます。

リストの値を含むグローバル変数

値の集合を含む索引付きグローバル変数の名前を指定します。グローバル変数内の各索引の項目がリストに作成されます。

共用 HATS のローカル・グローバル変数と共用グローバル変数は同じ名前を持つことができます。共用グローバル変数を使用してリスト項目にデータを取り込む場合は、このボックスを選択します。このボックスをクリアした場合は、ローカル・グローバル変数が使用されます。

リストの表題を含むグローバル変数

オプション。表題の集合を含む索引付きグローバル変数の名前を指定します。この値によって指定されるグローバル変数のサイズは、前の設定で指定されたサイズ以上である必要があります。この索引付きグローバル変数の索引は、グローバル変数の値の索引と一致する必要もあります (その結果、ユーザーに表示される実際の値と表題が同期します)。この値が指定されない場合は、リスト内の各項目の表題がその値になります。

ストリングから設定

これを選択した場合、指定したストリングがリストに設定されます。

リスト項目

オプション。リストに含まれる項目のストリングを指定します。項目はセミコロン (;) で分離する必要があります。リスト項目の表題とリスト項目の値が異なるようにするには、表題と値を等号 (=) で区切って両方を入力します。例えば、**Apple=A;Grape=G** の値は、2 つの項目「**Apple**」と「**Grape**」を持つリストをレンダリングします。最初の項目を選択すると、関連するホスト画面の入力フィールドに「**A**」が挿入されます。

リストの項目とホスト画面に挿入される値を同一にするには、項目のみを入力します。例えば、**Apple=A;G** です。この例では、リストとホスト画面の入力フィールドに **G** が表示されます。

ヒントから設定

これを選択した場合、コンポーネントに認識されたヒントがリストに設定されます (「ヒント付き入力フィールド」コンポーネントの使用時に限り適用可能)。

表題ソース (Caption source)

生成されるリストの表題を決定する方法を指定します。オプション: 「**コンポーネントから**」 (コンポーネントから取り出した表題を使用) または「**カスタム**」(指定された表題を使用)。

カスタム表題

オプション。リストの表題を指定します。

注: リストの表題を生成しない場合は、この設定はブランクのままにしてください。

表示可能な行数

ユーザーに対してリスト・ボックス内に表示される行の行数。リスト内の項

目数がこの数よりも多い場合は、スクロール・バーが表示されます。これにより、ユーザーはリスト全体をスクロールできます。デフォルトは 5 です。

1 行あたりの列数
次の行に折り返されるまでに水平方向に表示されるリストの数です。デフォルトは 1 です。

注:

1. この設定はデフォルト・レンダリングでは適用されません。
2. ウィジェットのプレビューは必ずしも全ページのプレビューと一致するとは限りません。これはウィジェットのプレビューが単にコンポーネントの出力をレンダリングしているだけであるために起こります。ウィジェットのプレビューは、全ページのプレビューで行われているように、レンダリングしたときに画面の配置を保存しようとはしません。

選択時に自動的に実行依頼
これを選択した場合、リストで選択操作が行われると、その選択内容が実行依頼されます。

送信ボタンを表示
これを選択した場合、リストの横に送信ボタンがレンダリングされます。

送信ボタンの表題
オプション。送信ボタンの表題を指定します。

リストのスタイル・クラス **Web のみ**
オプション。生成されたリストに関連する CSS スタイル・クラスです。HTML リスト・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSDROPDOWN** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

リスト・オプションのスタイル・クラス **Web のみ**
オプション。リストの各オプションに関連する CSS スタイル・クラスです。デフォルト値は **HATSOPTION** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

表題のスタイル・クラス **Web のみ**
オプション。生成されたリストの表題に関連する CSS スタイル・クラスです。デフォルト値は **HATSCAPTION** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス **Web のみ**
オプション。生成されるテーブルに関連する CSS スタイル・クラスです。複数のリストがレンダリングされる場合、HTML テーブルはこれらのリストを囲むように生成されます。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**
オプション。オーバーライドする任意の CSS プロパティー。例えば、このフィールドに「**font-color: red; font-size: 18pt;**」と指定して、このウィジェットのフォントの色およびサイズを変更できます。入力したプロパティーは、このウィジェットの各エレメントに適用されます。スタイル・プロパテ

ィー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド RCP のみ
デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド RCP のみ
前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

ポップアップ

ポップアップ・ウィジェットは、関連のランチャー・ボタン、リンク、またはイメージを持つ入力フィールドのレンダリングを担当します。このランチャーは、関連の入力フィールドの有効な値を選択するのに使用できるポップアップを開きます。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- コマンド行
- 入力フィールド
- ヒント付き入力フィールド
- 項目の選択

注:

1. プロキシー・サーバーを介して HATS アプリケーションがアクセスされたときにポップアップ・ウィジェット・イメージ・ランチャーを表示するには、プロキシー・サーバーを使用するよう HATS アプリケーションを構成する必要があります。手順については、 44 ページの『プロキシー・サーバーを使用するための HATS アプリケーションの構成』を参照してください。
2. このウィジェットを使用する際の DBCS に関する考慮事項については、 548 ページの『SBCS での最大長の除去』を参照してください。

次の図に、変換においてヒント付き入力フィールド・コンポーネントの例のデータを入力として使用するポップアップ・ウィジェットの表示例を示します。

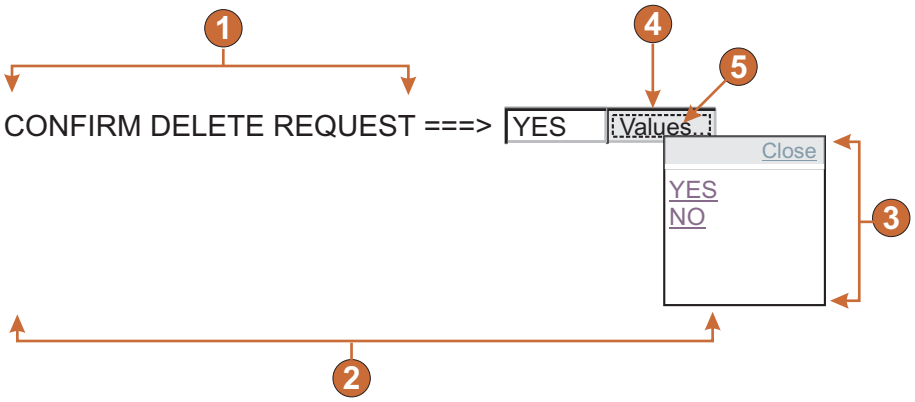


図 48. ポップアップ・ウィジェットの例

1. 表題 (コンポーネントから)
2. 1 行あたりの列数 (1)
3. 設定 (ヒントから)
4. ランチャー・タイプ (ボタン)
5. ランチャーの表題 (値)

以下の設定が、このウィジェットに対して構成できます。

グローバル変数から設定
これを選択した場合、指定したグローバル変数がポップアップ・リストに設定されます。

リストの値を含むグローバル変数
値の集合を含む索引付きグローバル変数の名前を指定します。グローバル変数内の各索引に対するリンクが、ポップアップ・ウィンドウに作成されます。

共用 HATS のローカル・グローバル変数と共用グローバル変数は同じ名前を持つことができます。共用グローバル変数を使用してリスト項目にデータを取り込む場合は、このボックスを選択します。このボックスをクリアした場合は、ローカル・グローバル変数が使用されます。

リストの表題を含むグローバル変数
オプション。表題の集合を含む索引付きグローバル変数の名前を指定します。この値によって指定されるグローバル変数のサイズは、前の設定で指定されたサイズ以上である必要があります。この索引付きグローバル変数の索引は、グローバル変数の値の索引と一致する必要もあります (その結果、ユーザーに表示される実際の値と表題が同期します)。この値が指定されない場合は、各リンクのテキストがその値になります。

ストリングから設定
これを選択した場合、指定したストリングがポップアップ・リストに設定されます。

リスト項目
オプション。ポップアップ・ウィンドウに含む項目のストリングを

指定します。項目はセミコロン (;) で分離する必要があります。リスト項目の表題とリスト項目の値が異なるようにするには、表題と値を等号 (=) で区切って両方を入力します。例えば、**Apple=A;Grape=G** の値は、2 つの項目「**Apple**」と「**Grape**」を持つポップアップ・ウィンドウをレンダリングします。最初の項目を選択すると、関連するホスト画面の入力フィールドに「**A**」が挿入されます。

ポップアップ・ウィンドウの項目とホスト画面に挿入される値を同一にするには、項目のみを入力します。例えば、**Apple=A;G** です。この例では、ポップアップ・ウィンドウとホスト画面の入力フィールドに **G** が表示されます。

ヒントから設定

これを選択した場合、コンポーネントによって認識されたヒントがポップアップにロードされます (「**ヒント付き入力フィールド**」コンポーネントの使用時に限り適用可能)。

表題ソース (Caption source)

生成される入力フィールドの表題を決定する方法を指定します。オプション: 「**コンポーネントから**」 (コンポーネントから取り出した表題を使用) または「**カスタム**」(指定された表題を使用)。

カスタム表題

オプション。入力フィールドの表題を指定します。注: 関連する入力フィールドの表題を生成しない場合は、この設定をブランクのままにしてください。

1 行あたりの列数

次の行に折り返されるまでに、水平方向に表示される入力フィールドの数です。

注:

1. この設定はデフォルト・レンダリングでは適用されません。
2. ウィジェットのプレビューは必ずしも全ページのプレビューと一致するとは限りません。これはウィジェットのプレビューが単にコンポーネントの出力をレンダリングしているだけであるために起こります。ウィジェットのプレビューは、全ページのプレビューで行われているように、レンダリングしたときに画面の配置を保存しようとはしません。

入力フィールドのカーソル位置オプションを使用可能にする Web のみ

このボックスを選択して、ユーザーが入力フィールドでデータ入力モードからカーソル位置決めモードに切り替えができるようにします。データ入力モードのときは、ユーザーは入力フィールドにデータを入力できます。カーソル位置決めモードのときは、タブ移動または別の方法で、入力フィールドの任意の文字上にカーソルを配置できます。この設定は他のカーソル位置決め機能を持たないモバイル装置用に最適化されている Web プロジェクトでは便利です。HATS Toolkit GUI に表示されないこのオプションの他の設定のついては、 54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

読み取り専用

これを選択した場合、レンダリングされた入力フィールドは読み取り専用に

なります。読み取り専用の入力フィールドは通常の入力フィールドのように表示されますが、ユーザーはこのフィールドの内容を変更できません。これは、無保護フィールドの内容をユーザーに対して表示するが、ユーザーがフィールドの内容を変更できないようにする場合に便利です。ポップアップ・ウィジェットの場合、フィールドに入力される入力データを制御できます。このフィールドには、ポップアップで使用可能な値のみを指定できます。

前景色を使用可能にする

これを選択した場合、ホスト画面の前景色がレンダリングされます。

Web アプリケーションの場合は、前景色はその色を表す CSS スタイルシート・クラスにマップされます。例えば、ホスト画面フィールドが **RED** とマークされている場合、フィールド・ウィジェットは、そのフィールド用に生成された HTML を、クラス名属性が **HRED** に設定されているタグで囲みます。この処理によって、ユーザーは、生成される Web ページ上にホスト画面の色を再マップすることができます。

注: blacktheme.css は、ホスト画面フィールドの前景色を直接マップします (例えば、ホスト画面上の赤色のテキストは、生成された Web ページ上で赤色のテキストとして表示されます)。ただし、monochrome.css などの他のスタイルシート・ファイルは、フィールドの色を別の方法でマップし、一貫性のある、最新のスタイルを作成しようとします。

リッチ・クライアント・アプリケーションの場合、色はリッチ・クライアント・テンプレートによってマップされます。

注: この設定は、「**色のオーバーライド**」>「**前景色**」の設定によってオーバーライドされます。

拡張属性を使用可能にする

これを選択した場合、拡張フィールド属性 (明滅、反転表示、下線、および列分離子) が表示されます。また、3270 Web アプリケーションの場合、拡張フィールドの色もマップされます (詳細については、前の設定を参照してください)。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

リッチ・クライアント・アプリケーションの場合、拡張フィールド属性はリッチ・クライアント・テンプレートによってマップされます。

明滅スタイル **Web のみ**

ホスト画面における明滅テキストのレンダリングに使用する CSS スタイル設定。

反転表示スタイル (Reverse video style) **Web のみ**

ホスト画面における反転表示テキストのレンダリングに使用する CSS スタイル設定。

下線スタイル (Underline style) **Web のみ**

ホスト画面における下線付きテキストのレンダリングに使用する CSS スタイル設定。

列分離子スタイル (Column separator style) **Web のみ**

ホスト画面における列分離子のレンダリングに使用する CSS スタイル設定。

ランチャー・タイプ
ランチャーのスタイルを指定します。このランチャーは、ポップアップをオープンするために使用します。オプション: 「**ボタン**」、「**リンク**」、または「**イメージ**」。

イメージ・ファイル名 (Image filename)
イメージ・ファイルの名前。これは、ランチャー・タイプが「イメージ」である場合にのみ使用可能です。イメージをプロジェクトにインポートするには、「**インポート**」をクリックします。プレビュー・ウィンドウに、現在選択しているイメージが表示されます。

表題
ランチャー・ボタンまたはリンクの表題。これは、ランチャー・タイプが「リンク」または「ボタン」である場合にのみ使用可能です。

リンクのスタイル・クラス Web のみ
オプション。生成されるランチャー・リンクに関連する CSS スタイル・クラスです。HTML リンク・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSLINK** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ボタンのスタイル・クラス Web のみ
オプション。生成されるランチャー・ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

リンクの表題を閉じる Web のみ
「クローズ」アクションで表示される表題。

入力フィールドのスタイル・クラス Web のみ
オプション。生成される入力フィールドに関連する CSS スタイル・クラスです。HTML 入力タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSINPUT** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

表題のスタイル・クラス Web のみ
オプション。生成される入力フィールドの表題に関連する CSS スタイル・クラスです。デフォルト値は **HATSCAPTION** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス Web のみ
オプション。生成されるテーブルに関連する CSS スタイル・クラスです。入力フィールドの複数の集合がレンダリングされる場合、1 つの HTML テーブルが生成され、それらの入力フィールドを囲みます。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ポップアップ・ウィンドウのスタイル・クラス Web のみ
オプション。生成されるポップアップ・ウィンドウに関連する CSS スタイ

ル・クラスです。デフォルト値は、**HATSPOPUP** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

クローズ・リンクのスタイル・クラス **Web のみ**
オプション。ポップアップ・ウィンドウのクローズに使用されるリンクに関連する CSS スタイル・クラスです。デフォルト値は **HATSLINK** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

項目リンクのスタイル・クラス **Web のみ**
オプション。ポップアップから項目を選択するために使用されるリンクに関連する CSS スタイル・クラスです。デフォルト値は **HATSPOPUPITEMLINK** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**
オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

注: 表示されるポップアップ・ウィジェットの透過性が高くなりすぎることがあります。ウィジェットの透過性を低くする場合は、以下の手順で、テンプレートが使用している CSS (カスケーディング・スタイル・シート) を変更する必要があります。

1. テンプレートが使用している CSS をオープンします。
2. **DIV.HATSPOPUP** クラスの下で、「filter: **alpha (opacity=40, style=0)**」を検索します。
3. opacity 値を変更してポップアップ・ウィジェットの透過性を調整します。**0** は透明、**100** は完全に不透明です。

フォントのオーバーライド **RCP のみ**
デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド **RCP のみ**
前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

ラジオ・ボタン (データ入力)

ラジオ・ボタン (データ入力) ウィジェットは、ラジオ・ボタン・セットをレンダリングします。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- コマンド行
- 入力フィールド

- ヒント付き入力フィールド

次の図に、変換においてヒント付き入力フィールド・コンポーネントの例のデータを入力として使用するラジオ・ボタン (データ入力) ウィジェットの表示例を示します。

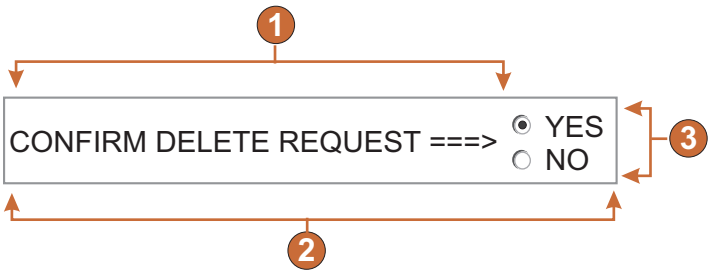


図 49. ラジオ・ボタン (データ入力) ウィジェットの例

1. 表題 (コンポーネントから)
2. 1 行あたりの列数 (1)
3. 設定 (ヒントから)

以下の設定が、このウィジェットに対して構成できます。

グローバル変数から設定
これを選択した場合、指定したグローバル変数がラジオ・ボタン・セットにロードされます。

リストの値を含むグローバル変数
値の集合を含む索引付きグローバル変数の名前を指定します。グローバル変数内の各索引に対するラジオ・ボタンが作成されます。

共用 HATS のローカル・グローバル変数と共用グローバル変数は同じ名前を持つことができます。共用グローバル変数を使用してリスト項目にデータを取り込む場合は、このボックスを選択します。このボックスをクリアした場合は、ローカル・グローバル変数が使用されます。

リストの表題を含むグローバル変数
オプション。表題の集合を含む索引付きグローバル変数の名前を指定します。この値によって指定されるグローバル変数のサイズは、前の設定で指定されたサイズ以上である必要があります。この索引付きグローバル変数の索引は、グローバル変数の値の索引と一致する必要もあります (その結果、ユーザーに表示される実際の値と表題が同期します)。この値が指定されない場合は、各ラジオ・ボタンの表題がその値になります。

ストリングから設定
これを選択した場合、指定したストリングがラジオ・ボタン・セットにロードされます。

リスト項目
オプション。ラジオ・ボタン・セットに含む項目のストリングを指定します。項目は**セミコロン (;)** で分離する必要があります。リス

ト項目の表題とリスト項目の値が異なるようにするには、表題と値を等号 (=) で区切って両方を入力します。例えば、値 **Apple=A;Grape=G** により、2 つの項目「**Apple**」と「**Grape**」のラジオ・ボタンがレンダリングされます。最初の項目を選択すると、関連するホスト画面の入力フィールドに「**A**」が挿入されます。

ラジオ・ボタンの集合の項目とホスト画面に挿入される値を同一にするには、項目のみを入力します。例えば、**Apple=A;G** です。この例では、ラジオ・ボタンの横、およびホスト画面の入力フィールドに、**G**が表示されます。

ヒントから設定

これを選択した場合、コンポーネントによって認識されたヒントがラジオ・ボタン・セットにロードされます (「ヒント付き入力フィールド」コンポーネントの使用時に限り適用可能)。

表題ソース (Caption source)

生成されるラジオ・ボタンの集合の表題を決定する方法を指定します。オプション: 「**コンポーネントから**」 (コンポーネントから取り出した表題を使用) または「**カスタム**」(指定された表題を使用)。

カスタム表題

オプション。集合の表題を指定します。

注: この集合の表題を生成しない場合、この設定はブランクのままにしてください。

表題のスペースをトリム (Trim spaces on caption)

これを選択した場合、表題の両端から空白 (余分なスペース) が除去されます。

1 行あたりの列数

次の行に折り返されるまでに、水平方向に表示されるラジオ・ボタンの数です。

ラベルのスタイル・クラス **Web のみ**

オプション。ラジオ・ボタンの集合に対して生成される表題に関連する CSS スタイル・クラスです。デフォルト値は **HATSCAPTION** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

ラジオ・ボタンのスタイル・クラス **Web のみ**

オプション。各ラジオ・ボタンに関連する CSS スタイル・クラスです。デフォルト値は **HATSRADIOBUTTON** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

ラジオ・ボタンの説明のスタイル・クラス **Web のみ**

オプション。各ラジオ・ボタンの説明に関連する CSS スタイル・クラスです。デフォルト値は **HATSTEXT** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス **Web のみ**

オプション。生成されるテーブルに関連する CSS スタイル・クラスです。複数のラジオ・ボタンがレンダリングされる場合は、1 つの HTML テーブルが生成され、それらのラジオ・ボタンを囲みます。HTML テーブル・タ

グの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**

オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド **RCP のみ**

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

ラジオ・ボタン (項目選択)

ラジオ・ボタン (項目選択) ウィジェットは、ラジオ・ボタン・セットをレンダリングします。

このウィジェットは、以下のコンポーネントの提供するデータをレンダリングします。

- 項目の選択

次の図に、変換において項目の選択コンポーネントの例のデータを入力として使用するラジオ・ボタン (項目選択) ウィジェットの表示例を示します。

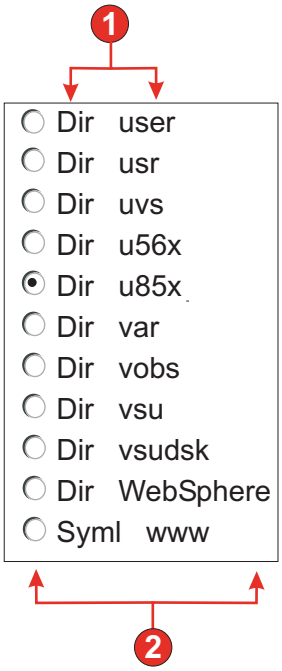


図 50. ラジオ・ボタン (項目選択) ウィジェットの例

1. 表題 (コンポーネントから)
2. 1 行あたりの列数 (デフォルトは 1)

以下の設定が、このウィジェットに対して構成できます。

表題　生成される入力フィールドの表題を決定する方法を指定します。オプション: 「**コンポーネントから**」 (コンポーネントから取り出した表題を使用) または「**カスタム**」(指定された表題を使用)。

カスタム表題
オプション。入力フィールドの表題を指定します。

注: チェック・ボックスの表題を生成しない場合は、この設定をブランクのままにしてください。

表題のスペースをトリム (Trim spaces on caption)
これを選択した場合、表題の両端から空白 (余分なスペース) が除去されます。

1 行あたりの列数
次の行に折り返されるまでに、水平方向に表示される入力フィールドの数です。

選択値 (Selection value)
値を選択したときにホスト画面の入力フィールドに入力される値。オプション: **/**、**S**、または **1**。

選択時に自動的に実行依頼
これを選択した場合、ラジオ・ボタンが選択されると、その項目選択が実行依頼されます。ラジオ・ボタンがファンクション・キーから生成された場合、選択された項目のキーがホストに送信されます。

送信ボタンを表示

これを選択した場合、ラジオ・ボタン・セットの下に送信ボタンがレンダリングされます。ファンクション・キー・コンポーネントとこのウィジェットを併用しているときには、適用しないでください。

送信ボタンの表題

オプション。送信ボタンの表題を指定します。

ボタンのスタイル・クラス **Web のみ**

オプション。生成される送信ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ラジオ・ボタンのスタイル・クラス **Web のみ**

オプション。各ラジオ・ボタンに関連する CSS スタイル・クラスです。デフォルト値は **HATSRADIOBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ラジオ・ボタンの説明のスタイル・クラス **Web のみ**

オプション。各ラジオ・ボタンの説明に関連する CSS スタイル・クラスです。デフォルト値は **HATSTEXT** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ラベルのスタイル・クラス **Web のみ**

オプション。ラジオ・ボタンの集合に対して生成される表題に関連する CSS スタイル・クラスです。デフォルト値は **HATSCAPTION** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス **Web のみ**

オプション。生成されるテーブルに関連する CSS スタイル・クラスです。複数のラジオ・ボタンがレンダリングされる場合は、1 つの HTML テーブルが生成され、それらのラジオ・ボタンを囲みます。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**

オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド **RCP のみ**

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

ラジオ・ボタン (選択)

ラジオ・ボタン (選択) ウィジェットは、ラジオ・ボタン・セットをレンダリングします。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- ファンクション・キー
- ライト・ペン (アテンション)
- 選択リスト
- 選択フィールド (ENPTUI)

次の図に、変換において選択リスト・コンポーネントの例のデータを入力として使用する選択リスト・ラジオ・ボタン・ウィジェットの表示例を示します。

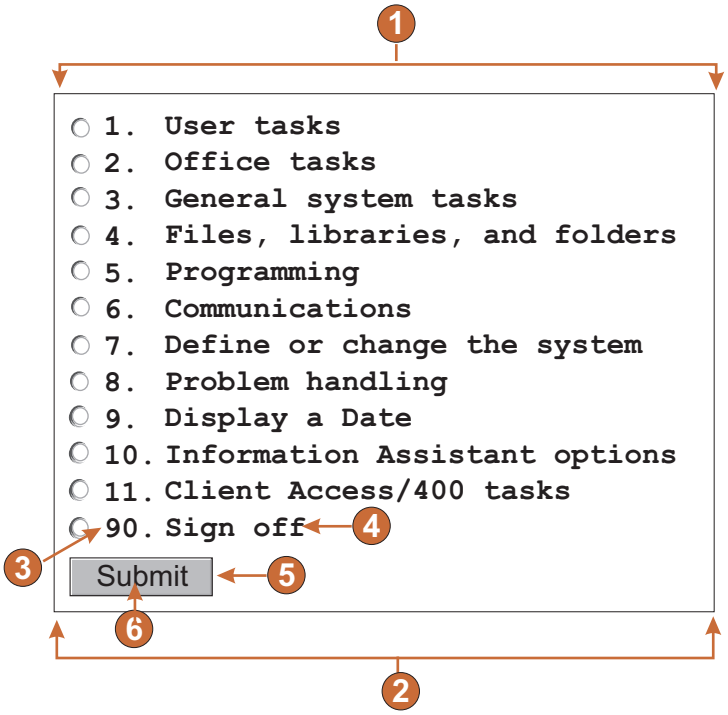


図 51. 選択リスト・ラジオ・ボタン・ウィジェットの例

1. 表題のタイプ (両方を表示)
2. 1 行あたりの列数 (1)
3. 先行トークン (両方を表示)
4. 説明 (両方を表示)
5. 送信ボタン (送信ボタンを表示)
6. 送信ボタンの表題 (送信)

以下の設定が、このウィジェットに対して構成できます。

表題のタイプ

各ボタンの表題を決定する方法を指定します。コンポーネントから先行トークンの値および説明が取り込まれます。つまり、ボタンの表題上に表示され

る項目を選択できます。例えば、ホスト画面に「**4. メール**」というメニュー項目が存在する場合、表題表示として「**4**」、「**メール**」、または「**4. メール**」を選択できます。

表題のスペースをトリム (Trim spaces on caption)
これを選択した場合、表題の両端から空白 (余分なスペース) が除去されます。

1 行あたりの列数
次の行に折り返されるまでに、水平方向に表示されるラジオ・ボタンの数です。

選択時に自動的に実行依頼
これを選択した場合、ラジオ・ボタンが選択されると、選択が実行依頼されます。ラジオ・ボタンがファンクション・キーから生成された場合、選択された項目のキーがホストに送信されます。ラジオ・ボタンが選択リストから生成された場合、選択された項目の値が、ターゲット・フィールドに挿入され、指定されたアクション・キーがホストに送信されます。

送信ボタンを表示
これを選択した場合、ラジオ・ボタン・セットの下に送信ボタンがレンダリングされます。ファンクション・キー・コンポーネントとこのウィジェットを併用しているときには、適用しないでください。

送信ボタンの表題
オプション。送信ボタンの表題を指定します。

ボタンのスタイル・クラス **Web のみ**
オプション。生成される送信ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ラジオ・ボタンのスタイル・クラス **Web のみ**
オプション。各ラジオ・ボタンに関連する CSS スタイル・クラスです。デフォルト値は **HATSRADIOBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ラジオ・ボタンの説明のスタイル・クラス **Web のみ**
オプション。各ラジオ・ボタンの説明に関連する CSS スタイル・クラスです。デフォルト値は **HATSTEXT** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ラベルのスタイル・クラス **Web のみ**
オプション。ラジオ・ボタンの集合に対して生成される表題に関連する CSS スタイル・クラスです。デフォルト値は **HATSCAPTION** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス **Web のみ**
オプション。生成されるテーブルに関連する CSS スタイル・クラスです。複数のラジオ・ボタンがレンダリングされる場合は、1 つの HTML テーブルが生成され、それらのラジオ・ボタンを囲みます。HTML テーブル・タ

グの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル Web のみ
オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド RCP のみ
デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

スクロール・バー (ENPTUI)

スクロール・バー (ENPTUI) ウィジェットは、スクロール・バー・フィールド (ENPTUI) コンポーネントから提供されたデータをレンダリングします。

次の図に、変換においてスクロール・バー・フィールド (ENPTUI) コンポーネントの例のデータを入力として使用するスクロール・バー (ENPTUI) ウィジェットの表示例を示します。

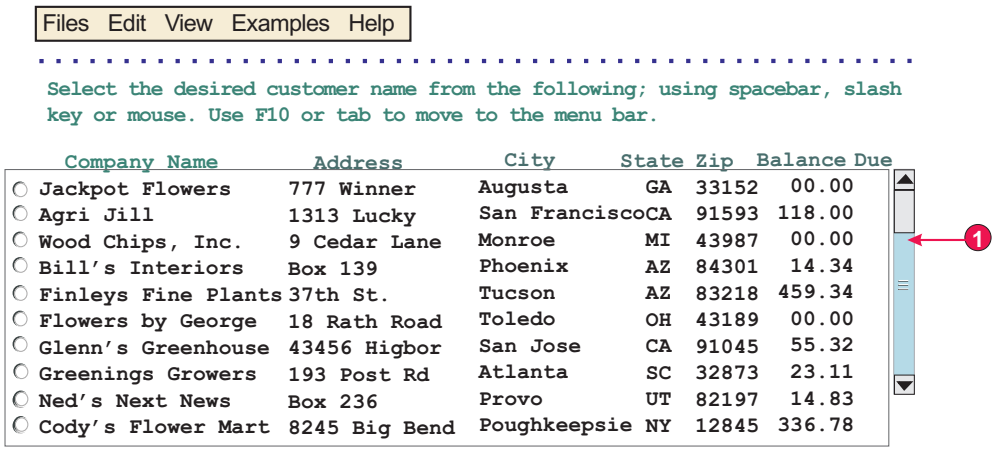


図 52. スクロール・バー (ENPTUI) ウィジェットの例

1. スクロール・バー

以下の設定が、このウィジェットに対して構成できます。

スクロール・バーのスタイル・クラス Web のみ
オプション。スクロール・バーの外観に関連する CSS スタイル・クラスです。デフォルト値は **HATSESCROLL** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スクロール・バー・ボタンのスタイル・クラス **Web のみ**
オプション。スクロール・バー・ボタンに関連する CSS スタイル・クラスです。デフォルト値は **HATSESCROLLBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ボタン矢印の色 **Web のみ**
ボタン矢印の色です。デフォルト値は 「**黒**」 です。

スタイル **Web のみ**
オプション。オーバーライドする任意の CSS プロパティー。例えば、このフィールドに「**font-color: red; font-size: 18pt;**」と指定して、このウィジェットのフォントの色およびサイズを変更できます。入力したプロパティーは、このウィジェットの各エレメントに適用されます。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

色のオーバーライド **RCP のみ**
背景色をオーバーライドするには、このオプションを選択し、提供される「**背景色**」設定コントロールを使用します。

サブファイル (チェック・ボックス)

サブファイル (チェック・ボックス) ウィジェットは、5250 サブファイルをレンダリングします。

このウィジェットは、サブファイル・コンポーネントが提供するデータをレンダリングします。

注: このウィジェットを使用する際の DBCS に関する考慮事項については、548 ページの『SBCS での最大長の除去』を参照してください。

次の図に、変換においてサブファイル・コンポーネントの例のデータを入力として使用するサブファイル (チェック・ボックス) ウィジェットの表示例を示します。

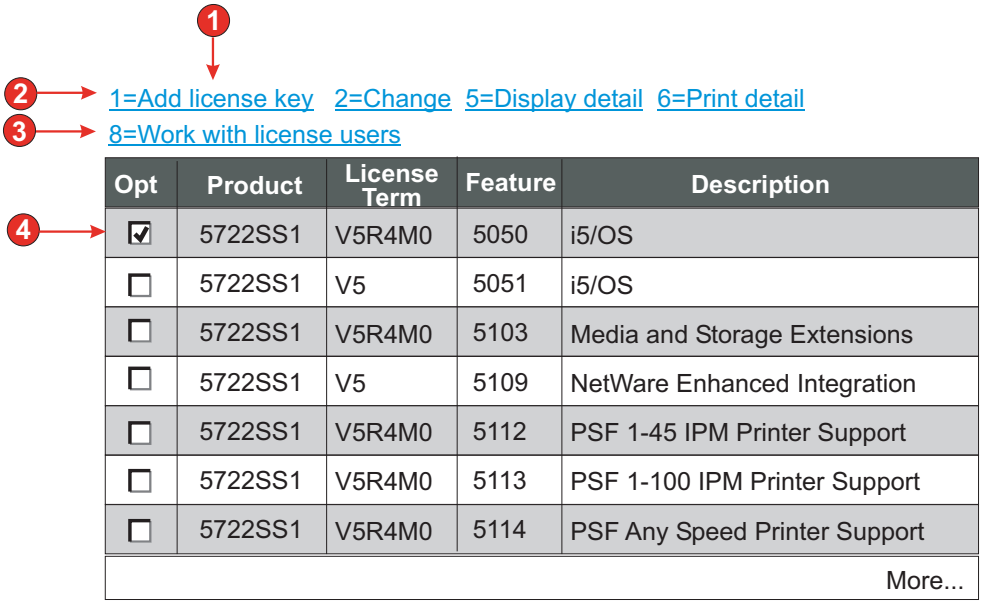


図 53. サブファイル (チェック・ボックス) ウィジェットの例

1. 表題のタイプ (両方を表示)
2. アクション・タイプ (リンク)
3. アクションの場所 (サブファイルの上)
4. チェック・ボックス

以下の設定が、このウィジェットに対して構成できます。

表題のタイプ:
各チェック・ボックスの表題を決定する方法を指定します。コンポーネントから先行トークンの値および説明が取り込まれます。つまり、表題として表示する項目を選択できます。例えば、ホスト画面に「**5. 表示**」というアクションが存在する場合、表題表示として「**5**」、「**表示**」、または「**5. 表示**」を選択できます。

アクション・タイプ
アクションをトリガーするために「**ボタン**」または「**リンク**」のいずれを使用するかを指定します。

アクションの場所
アクションをトリガーするボタンまたはリンクの位置を指定します。オプションは「**サブファイルの上と下**」、「**サブファイルの上**」、および「**サブファイルの下**」です。

1 レコードあたりの行数 (Rows per record)

行の代替色の表示 RCP のみ
行を代替色でレンダリングするには、このボックスを選択します。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター (http://www.ibm.com/support/knowledgecenter/

SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

行数 (Number of rows)
サブファイル内の 1 レコードあたりの固定行数を指定します。

列の配置 **Web のみ**

このウィジェットは、データの基本列を表示し、必要に応じて追加詳細列を表示する詳細セクションを持つように構成できます。この設定は、モバイル装置にテーブル・データを表示するときに便利です。これらの設定および HATS Toolkit GUI に表示されないこのオプションの他の設定について詳しくは、54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

基本列 このフィールドを使用して、表示する列およびその順序を指定します。フィールドでは、拡張簡略記述を使用して、列番号の数値での任意の順序でリスト表示をすることができ、列を複数回表示したり、またはまったく表示させないこともできます (例: 「**1,3,2,5-7,2-3,9**」)。

2 つの数字の間のハイフン (-) は列の範囲を示します。アスタリスク文字 (*) を使用して、使用可能な最後の列を示すことができます。アスタリスクのみは、列 1 から n (n は使用可能な最後の列) を意味します。また、5-* の省略形として 5* を使用することもできます。1-* と * は同じ意味です。

Web プロジェクトでは、この設定のデフォルトは *****、つまりすべての列になります。モバイル装置に最適化された Web プロジェクトでは、この設定のデフォルトは **1-2**、つまり列 1 と列 2 になっています。

注: 数値はコンポーネントによってウィジェットに提供された列を表します。ウィジェットに提供される前に、除外されるかどうかわからないコンポーネントによって最初に認識された列を表すものではありません。

詳細列 このオプションのフィールドを使用して、要求されたときに表示する詳細列およびその順序を指定します。フィールドでは、拡張簡略記述を使用して、数値での任意の順序でリスト表示することができます。

Web プロジェクトでは、この設定のデフォルトは空、つまり詳細列セクションはありません。モバイル装置に最適化された Web プロジェクトでは、この設定のデフォルトは **3***、つまり列 3 から最後の列になっています。

詳細列をサーバーに保持
このオプションを選択して、ユーザーによって要求されるまで詳細列セクションをサーバーに保持します。これによって、初期レンダリングでの HTML 出力の量を削減できます。デフォルトはクリアされます。

強調表示された行 RCP のみ

レンダリング時に、強調表示する行の集合を、コンマで区切って指定するか、または範囲指定します。この設定は、生成されたテーブル内の重要な行を強調表示する場合に便利です。この値で表される行は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

強調表示された列 RCP のみ

レンダリング時に強調表示する列の集合を、コンマで区切って指定するか、または範囲指定します。この設定は、生成されたテーブル内の重要な列を強調表示する場合に便利です。この値で表される列は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

格子線の表示 RCP のみ

レンダリングされるサブファイルで格子線を表示するには、このボックスを選択します。

列見出しのソース RCP のみ

列ヘッダーのテキストを、サブファイル・ヘッダー・コンポーネントから取り込むか (「**コンポーネントから**」)、またはユーザーが指定するか (「**カスタム**」) を、ドロップダウンから選択します。サブファイル・ヘッダー・コンポーネントから取り込む場合は、詳細について 248 ページの『ヘッダー』のコンポーネントの説明を参照してください。

列見出し (コンマ区切り) RCP のみ

「**列見出しのソース**」を「**カスタム**」に設定した場合に、複数の列ヘッダーをコンマで区切ってここに入力します。

クラシック見出しスタイルを使用 RCP のみ

クラシック見出しスタイルを表示させるには、このボックスを選択します。また、「**列見出しのソース**」が「**コンポーネントから**」に設定されており、サブファイル・ヘッダー・コンポーネントにより複数行ヘッダーが検出されると、複数行ヘッダーが表示されます。デフォルトの SWT テーブル制御ヘッダーを単一行でのみ表示するには、このボックスをクリアします。

ヘッダーのスペースをトリム (Trim spaces on headers) RCP のみ

これを選択した場合、ヘッダー行のセルの内部にあるテキストの両端から空白 (余分なスペース) が除去されます。

前景色を使用可能にする

これを選択した場合、ホスト画面の前景色がレンダリングされます。

Web アプリケーションの場合は、前景色はその色を表す CSS スタイルシート・クラスにマップされます。例えば、ホスト画面フィールドが **RED** とマークされている場合、フィールド・ウィジェットは、そのフィールド用に生成された HTML を、クラス名属性が **HRED** に設定されているタグで囲みます。この処理によって、ユーザーは、生成される Web ページ上にホスト画面の色を再マップすることができます。

注: blacktheme.css は、ホスト画面フィールドの前景色を直接マップします (例えば、ホスト画面上の赤色のテキストは、生成された Web ページ上で赤色のテキストとして表示されます)。ただし、monochrome.css などの他のスタイルシート・ファイルは、フィールドの色を別の方法でマップし、一貫性のある、最新のスタイルを作成しようとします。

リッチ・クライアント・アプリケーションの場合、色はリッチ・クライアント・テンプレートによってマップされます。

注: この設定は、「**色のオーバーライド**」>「**前景色**」の設定によってオーバーライドされます。

フィールドのスタイル・クラス **Web のみ**
オプション。生成されたフィールドに関連する CSS スタイル・クラスです。HTML タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSFIELD** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

拡張属性を使用可能にする
これを選択した場合、拡張フィールド属性 (明滅、反転表示、下線、および列分離子) が指定されたスタイルで表示されます。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

明滅スタイル **Web のみ**
ホスト画面における明滅テキストのレンダリングに使用する CSS スタイル設定。

反転表示スタイル (Reverse video style) **Web のみ**
ホスト画面における反転表示テキストのレンダリングに使用する CSS スタイル設定。

下線スタイル (Underline style) **Web のみ**
ホスト画面における下線付きテキストのレンダリングに使用する CSS スタイル設定。

列分離子スタイル (Column separator style) **Web のみ**
ホスト画面における列分離子のレンダリングに使用する CSS スタイル設定。

入力フィールドの下線文字を除去 (Strip underscores on input field)
テキストをレンダリングするときにテキストから下線文字を除去する場合は、このボックスを選択します。

入力フィールドのスペースをトリム
入力フィールドから先行および後続のスペースをトリムするには、このボックスを選択します。

入力フィールドのカーソル位置オプションを使用可能にする **Web のみ**

このボックスを選択して、ユーザーが入力フィールドでデータ入力モードからカーソル位置決めモードに切り替えができるようにします。データ入力モードのときは、ユーザーは入力フィールドにデータを入力できます。カーソル位置決めモードのときは、タブ移動または別の方法で、入力フィールドの任意の文字上にカーソルを配置できます。この設定は他のカーソル位置決め機能を持たないモバイル装置用に最適化されている Web プロジェクトでは便利です。HATS Toolkit GUI に表示されないこのオプションの他の設定のついては、54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

スプレッドシート・ファイルの提供

これを選択した場合、テーブル・データをスプレッドシート・ファイル・フォーマットで取得するためのランチャーが表示されます。

ファイル名の接頭部

スプレッドシート・ファイル名の接頭部を指定します。デフォルトは **spreadsheet** です。完全なファイル名の形式は `FileNamePrefix_TimeStamp` です。タイム・スタンプの形式は MMDDYY_HHMMSS (例: spreadsheet_041506_113020) です。

ファイル名の拡張子

スプレッドシート・フォーマット・タイプの拡張子名。オプションは **.csv** および **.xls** です。.csv ファイルは、データ列がコンマによって区切られている完全なテキスト・ファイルです。.xls ファイルは、Microsoft Excel Biff3 ファイルです。デフォルトは **.csv** です。

ランチャー・タイプ

ユーザーに対して表示されるランチャーのタイプ。オプションは、「**リンク**」、「**イメージ**」、および「**ボタン**」です。デフォルトは、「**リンク**」(Web プロジェクト) または「**ボタン**」(リッチ・クライアント プロジェクト) です。

ランチャーの表題

ユーザーに対して表示されるランチャーの表題。デフォルトは、「**ダウンロード**」(Web プロジェクト) または「**エクスポート**」(リッチ・クライアント・プロジェクト) です。

ボタンのスタイル・クラス **Web のみ**

生成されるランチャー・ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

リンクのスタイル・クラス **Web のみ**

オプション。生成されるランチャー・リンクに関連する CSS スタイル・クラスです。HTML リンク・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSLINK** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

リンクのスタイル・クラス **Web のみ**

オプション。生成されたアクション・リンクに関連する CSS スタイル・クラスです。HTML リンク・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSLINK** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

ボタンのスタイル・クラス **Web のみ**

オプション。生成されたアクション・ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

チェック・ボックスのスタイル・クラス **Web のみ**

オプション。生成されるチェック・ボックスに関連する CSS スタイル・クラスです。HTML チェック・ボックス・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSCHECKBOX** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス **Web のみ**

オプション。生成されるテーブルに関連する CSS スタイル・クラスです。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

テーブル・セルのスタイル・クラス **Web のみ**

オプション。生成されるテーブル内の各セルに関連する CSS スタイル・クラスです。HTML テーブル・セル (td) タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLECELL** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

見出し行のスタイル・クラス **Web のみ**

オプション。生成されるテーブル内の各ヘッダー行または列に関連する CSS スタイル・クラスです。HTML テーブルの行またはセル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLEHEADER** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

奇数行のスタイル・クラス **Web のみ**

オプション。テーブル内の奇数番の各行に関連する CSS スタイル・クラスです。奇数行番号の各 HTML テーブル行タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLEODDROW** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

偶数行のスタイル・クラス **Web のみ**

オプション。テーブル内の偶数番の各行に関連する CSS スタイル・クラスです。偶数行番号の各 HTML テーブル行タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLEEVENROW** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**

オプション。スタイル・プロパティー・ダイアログを表示するには、このフ

ィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド **RCP のみ**

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド **RCP のみ**

前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

注: リッチ・クライアント・プロジェクトでは、このウィジェットの各種色設定が次の順序 (最も高い優先度から低い優先度への順序) で適用されます。

1. 強調表示された行/列
2. 色のオーバーライド
3. 拡張属性を使用可能にする/前景色を使用可能にする
4. 行の代替色の表示

fieldTextAlignment **Web のみ**

Web プロジェクトで Sans Serif などの非モノスペース・フォントを使用すると、レンダリングされたデータはワード間のスペーシングの問題で適切には位置合わせされません。ソース設定 fieldTextAlignment を使用して、データの位置合わせを修正できます。この設定をプロジェクト全体で有効にするには、application.hap ファイルのソース・ビューを開き、SubfileCheckboxWidget ウィジェットのクラスを見つけ、次の例に示すように、設定値を NO_SEGMENTING に変更します。

```
<class name="com.ibm.hats.transform.widgets.SubfileCheckboxWidget">
.
   <setting name="fieldTextAlignment" value="NO_SEGMENTING"/>
.
</class>
```

この設定を変換内のコンポーネント・レベルで有効にするには、変換のソース・ビューを開き、SubfileCheckboxWidget ウィジェットの HATS:Component タグを見つけ、次の例に示すように、設定値を NO_SEGMENTING に変更します。

```
...
widget="com.ibm.hats.transform.widgets.SubfileCheckboxWidget"
...
widgetSettings="...|fieldTextAlignment:NO_SEGMENTING|..."
...
```

この設定の値は、WORD_BY_WORD および NO_SEGMENTING です。デフォルト値は **WORD_BY_WORD** です。

注: サブファイル・コンポーネントでデータ認識に「テーブル (可視)」認識オプションを使用している場合、この設定は効果がありません。

サブファイル (ドロップダウン)

サブファイル (ドロップダウン) ウィジェットは、5250 サブファイルをレンダリングします。

このウィジェットは、サブファイル・コンポーネントが提供するデータをレンダリングします。

注: このウィジェットを使用する際の DBCS に関する考慮事項については、548 ページの『SBCS での最大長の除去』を参照してください。

次の図に、変換においてサブファイル・コンポーネントの例のデータを入力として使用するサブファイル (ドロップダウン) ウィジェットの表示例を示します。

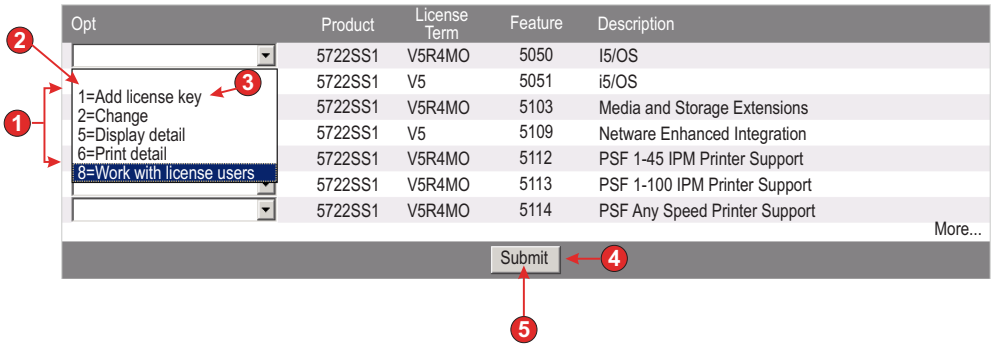


図 54. サブファイル (ドロップダウン) ウィジェットの例

1. 表題のタイプ (両方を表示)
2. 先行トークン
3. Description
4. 送信ボタン (送信ボタンを表示)
5. 送信ボタンの表題 (送信)

以下の設定が、このウィジェットに対して構成できます。

表題のタイプ:

ドロップダウン内の各項目の表題を決定する方法を指定します。コンポーネントから先行トークンの値および説明が取り込まれます。つまり、表題として表示する項目を選択できます。例えば、ホスト画面に「**5. 表示**」というアクションが存在する場合、表題表示として「**5**」、「**表示**」、または「**5. 表示**」を選択できます。

サブファイル・アクションを表示 (Show subfile actions)

ドロップダウン・リストに加え、レンダリングされたサブファイルとサブファイル・アクションの説明文 (例: **2=Add**) を表示する場合は、このボックスを選択します。このチェック・ボックスは、「**説明を表示**」を選択した場合には使用できません。

選択時に自動的に実行依頼

これを選択した場合、ドロップダウンで選択操作が行われると、その選択内容が実行依頼されます。

送信ボタンを表示

これを選択した場合、サブファイルの下に送信ボタンが表示されます。

送信ボタンの表題

オプション。送信ボタンの表題を指定します。

ボタンのスタイル・クラス **Web のみ**

オプション。生成される送信ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ボタン行のスタイル・クラス **Web のみ**

オプション。送信ボタンを含む生成されるテーブル行に関連する CSS スタイル・クラスです。デフォルト値は **HATSTABLEHEADER** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

1 レコードあたりの行数 (Rows per record)

行の代替色の表示 **RCP のみ**

行を代替色でレンダリングするには、このボックスを選択します。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

行数 (Number of rows)

サブファイル内の 1 レコードあたりの固定行数を指定します。

列の配置 **Web のみ**

このウィジェットは、データの基本列を表示し、必要に応じて追加詳細列を表示する詳細セクションを持つように構成できます。この設定は、モバイル装置にテーブル・データを表示するときに便利です。これらの設定および HATS Toolkit GUI に表示されないこのオプションの他の設定について詳しくは、54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

基本列 このフィールドを使用して、表示する列およびその順序を指定します。フィールドでは、拡張簡略記述を使用して、列番号の数値での任意の順序でリスト表示をすることができ、列を複数回表示したり、またはまったく表示させないこともできます (例: 「**1,3,2,5-7,2-3,9**」)。

2 つの数字の間のハイフン (-) は列の範囲を示します。アスタリスク文字 (*) を使用して、使用可能な最後の列を示すことができます。アスタリスクのみは、列 1 から n (n は使用可能な最後の列) を意味します。また、5-* の省略形として 5* を使用することもできます。1-* と * は同じ意味です。

Web プロジェクトでは、この設定のデフォルトは *****、つまりすべての列になります。モバイル装置に最適化された Web プロジェクトでは、この設定のデフォルトは **1-2**、つまり列 1 と列 2 になっています。

注: 数値はコンポーネントによってウィジェットに提供された列を表します。ウィジェットに提供される前に、除外されるかどうかわからないコンポーネントによって最初に認識された列を表すものではありません。

詳細列 このオプションのフィールドを使用して、要求されたときに表示する詳細列およびその順序を指定します。フィールドでは、拡張簡略記述を使用して、数値での任意の順序でリスト表示することができます。

Web プロジェクトでは、この設定のデフォルトは空、つまり詳細列セクションはありません。モバイル装置に最適化された Web プロジェクトでは、この設定のデフォルトは **3***、つまり列 3 から最後の列になっています。

詳細列をサーバーに保持

このオプションを選択して、ユーザーによって要求されるまで詳細列セクションをサーバーに保持します。これによって、初期レンダリングでの HTML 出力の量を削減できます。デフォルトはクリアされます。

強調表示された行 **RCP のみ**

レンダリング時に、強調表示する行の集合を、コンマで区切って指定するか、または範囲指定します。この設定は、生成されたテーブル内の重要な行を強調表示する場合に便利です。この値で表される行は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

強調表示された列 **RCP のみ**

レンダリング時に強調表示する列の集合を、コンマで区切って指定するか、または範囲指定します。この設定は、生成されたテーブル内の重要な列を強調表示する場合に便利です。この値で表される列は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

格子線の表示 RCP のみ

レンダリングされるサブファイルで格子線を表示するには、このボックスを選択します。

列見出しのソース RCP のみ

列ヘッダーのテキストを、サブファイル・ヘッダー・コンポーネントから取り込むか (「**コンポーネントから**」)、またはユーザーが指定するか (「**カスタム**」) を、ドロップダウンから選択します。サブファイル・ヘッダー・コンポーネントから取り込む場合は、詳細について 248 ページの『ヘッダー』のコンポーネントの説明を参照してください。

列見出し (コンマ区切り) RCP のみ

「**列見出しのソース**」を「**カスタム**」に設定した場合に、複数の列ヘッダーをコンマで区切ってここに入力します。

クラシック見出しスタイルを使用 RCP のみ

クラシック見出しスタイルを表示させるには、このボックスを選択します。また、「**列見出しのソース**」が「**コンポーネントから**」に設定されており、サブファイル・ヘッダー・コンポーネントにより複数行ヘッダーが検出されると、複数行ヘッダーが表示されます。デフォルトの SWT テーブル制御ヘッダーを単一行でのみ表示するには、このボックスをクリアします。

ヘッダーのスペースをトリム (Trim spaces on headers) RCP のみ

これを選択した場合、ヘッダー行のセルの内部にあるテキストの両端から空白 (余分なスペース) が除去されます。

前景色を使用可能にする

これを選択した場合、ホスト画面の前景色がレンダリングされます。

Web アプリケーションの場合は、前景色はその色を表す CSS スタイルシート・クラスにマップされます。例えば、ホスト画面フィールドが **RED** とマークされている場合、フィールド・ウィジェットは、そのフィールド用に生成された HTML を、クラス名属性が **HRED** に設定されているタグで囲みます。この処理によって、ユーザーは、生成される Web ページ上にホスト画面の色を再マップすることができます。

注: blacktheme.css は、ホスト画面フィールドの前景色を直接マップします (例えば、ホスト画面上の赤色のテキストは、生成された Web ページ上で赤色のテキストとして表示されます)。ただし、monochrome.css などの他のスタイルシート・ファイルは、フィールドの色を別の方法でマップし、一貫性のある、最新のスタイルを作成しようとします。

リッチ・クライアント・アプリケーションの場合、色はリッチ・クライアント・テンプレートによってマップされます。

注: この設定は、「**色のオーバーライド**」>「**前景色**」の設定によってオーバーライドされます。

フィールドのスタイル・クラス Web のみ

オプション。生成されたフィールドに関連する CSS スタイル・クラスです。HTML タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSFIELD** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

拡張属性を使用可能にする
これを選択した場合、拡張フィールド属性 (明滅、反転表示、下線、および列分離子) が指定されたスタイルで表示されます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

明滅スタイル **Web のみ**
ホスト画面における明滅テキストのレンダリングに使用する CSS スタイル設定。

反転表示スタイル (Reverse video style) **Web のみ**
ホスト画面における反転表示テキストのレンダリングに使用する CSS スタイル設定。

下線スタイル (Underline style) **Web のみ**
ホスト画面における下線付きテキストのレンダリングに使用する CSS スタイル設定。

列分離子スタイル (Column separator style) **Web のみ**
ホスト画面における列分離子のレンダリングに使用する CSS スタイル設定。

入力フィールドの下線文字を除去 (Strip underscores on input field)
テキストをレンダリングするときにテキストから下線文字を除去する場合は、このボックスを選択します。

入力フィールドのスペースをトリム
入力フィールドから先行および後続のスペースをトリムするには、このボックスを選択します。

入力フィールドのカーソル位置オプションを使用可能にする **Web のみ**
このボックスを選択して、ユーザーが入力フィールドでデータ入力モードからカーソル位置決めモードに切り替えができるようにします。データ入力モードのときは、ユーザーは入力フィールドにデータを入力できます。カーソル位置決めモードのときは、タブ移動または別の方法で、入力フィールドの任意の文字上にカーソルを配置できます。この設定は他のカーソル位置決め機能を持たないモバイル装置用に最適化されている Web プロジェクトでは便利です。HATS Toolkit GUI に表示されないこのオプションの他の設定のついては、 54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

スプレッドシート・ファイルの提供
これを選択した場合、テーブル・データをスプレッドシート・ファイル・フォーマットで取得するためのランチャーが表示されます。

ファイル名の接頭部
スプレッドシート・ファイル名の接頭部を指定します。デフォルトは **spreadsheet** です。完全なファイル名の形式は `FileNamePrefix_TimeStamp` です。タイム・スタンプの形式は MMDDYY_HHMMSS (例: spreadsheet_041506_113020) です。

ファイル名の拡張子
スプレッドシート・フォーマット・タイプの拡張子名。オプションは **.csv** および **.xls** です。.csv ファイルは、データ列がコンマによ

って区切られている完全なテキスト・ファイルです。.xls ファイルは、Microsoft Excel Biff3 ファイルです。デフォルトは **.csv** です。

ランチャー・タイプ
ユーザーに対して表示されるランチャーのタイプ。オプションは、「**リンク**」、「**イメージ**」、および「**ボタン**」です。デフォルトは、「**リンク**」(Web プロジェクト) または「**ボタン**」(リッチ・クライアント プロジェクト) です。

ランチャーの表題
ユーザーに対して表示されるランチャーの表題。デフォルトは、「**ダウンロード**」(Web プロジェクト) または「**エクスポート**」(リッチ・クライアント・プロジェクト) です。

ボタンのスタイル・クラス **Web のみ**
生成されるランチャー・ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

リンクのスタイル・クラス **Web のみ**
オプション。生成されるランチャー・リンクに関連する CSS スタイル・クラスです。HTML リンク・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSLINK** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ドロップダウンのスタイル・クラス **Web のみ**
オプション。生成されるドロップダウンに関連する CSS スタイル・クラスです。HTML ドロップダウン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSDROPDOWN** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

リスト・オプションのスタイル・クラス **Web のみ**
オプション。ドロップダウンの各オプションに関連する CSS スタイル・クラスです。デフォルト値は **HATSOPTION** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス **Web のみ**
オプション。生成されるテーブルに関連する CSS スタイル・クラスです。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブル・セルのスタイル・クラス **Web のみ**
オプション。生成されるテーブル内の各セルに関連する CSS スタイル・クラスです。HTML テーブル・セル (td) タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLECELL** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

見出し行のスタイル・クラス **Web のみ**
オプション。生成されるテーブル内の各ヘッダー行または列に関連する CSS スタイル・クラスです。HTML テーブルの行またはセル・タグの class

属性の値には、この値がセットされます。デフォルト値は **HATSTABLEHEADER** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

奇数行のスタイル・クラス **Web のみ**

オプション。テーブル内の奇数番の各行に関連する CSS スタイル・クラスです。奇数行番号の各 HTML テーブル行タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLEODDROW** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

偶数行のスタイル・クラス **Web のみ**

オプション。テーブル内の偶数番の各行に関連する CSS スタイル・クラスです。偶数行番号の各 HTML テーブル行タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLEEVENROW** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**

オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド **RCP のみ**

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド **RCP のみ**

前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

注: リッチ・クライアント・プロジェクトでは、このウィジェットの各種色設定が次の順序 (最も高い優先度から低い優先度への順序) で適用されます。

1. 強調表示された行/列
2. 色のオーバーライド
3. 拡張属性を使用可能にする/前景色を使用可能にする
4. 行の代替色の表示

fieldTextAlignment **Web のみ**

Web プロジェクトで Sans Serif などの非モノスペース・フォントを使用すると、レンダリングされたデータはワード間のスペーシングの問題で適切には位置合わせされません。ソース設定 fieldTextAlignment を使用して、データの位置合わせを修正できます。この設定をプロジェクト全体で有効にするには、application.hap ファイルのソース・ビューを開き、SubfileWidgetV6 ウィジェットのクラスを見つけ、次の例に示すように、設定値を `NO_SEGMENTING` に変更します。

```
<class name="com.ibm.hats.transform.widgets.SubfileWidgetV6">
.
   <setting name="fieldTextAlignment" value="NO_SEGMENTING"/>
.
</class>
```

この設定を変換内のコンポーネント・レベルで有効にするには、変換のソース・ビューを開き、SubfileWidgetV6 ウィジェットの HATS:Component タグを見つけ、次の例に示すように、設定値を NO_SEGMENTING に変更します。

```
...
widget="com.ibm.hats.transform.widgets.SubfileWidgetV6"
...
widgetSettings="...|fieldTextAlignment:NO_SEGMENTING|..."
...
```

この設定の値は、WORD_BY_WORD および NO_SEGMENTING です。デフォルト値は **WORD_BY_WORD** です。

注: サブファイル・コンポーネントでデータ認識に「テーブル (可視)」認識オプションを使用している場合、この設定は効果がありません。

サブファイル (ポップアップ)

サブファイル (ポップアップ) ウィジェットは、5250 サブファイルをレンダリングします。

このウィジェットは、サブファイル・コンポーネントが提供するデータをレンダリングします。

注: このウィジェットを使用する際の DBCS に関する考慮事項については、 548 ページの『SBCS での最大長の除去』を参照してください。

次の図に、変換においてサブファイル・コンポーネントの例のデータを入力として使用するサブファイル・ウィジェットの表示例を示します。

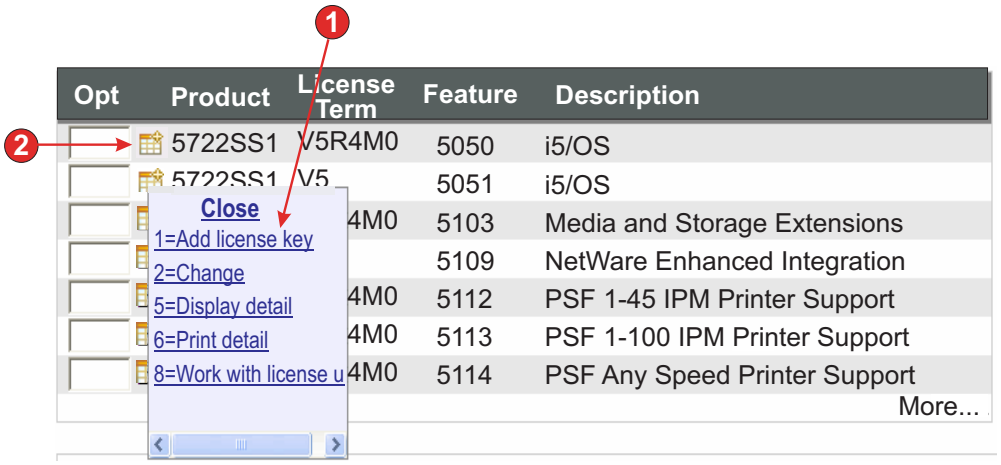


図 55. サブファイル (ポップアップ) ウィジェットの例

1. 表題のタイプ (両方を表示)
2. ランチャー・タイプ (イメージ)

以下の設定が、このウィジェットに対して構成できます。

表題のタイプ:
ポップアップ・ダイアログの各項目の表題を決定する方法を指定します。コンポーネントから先行トークンの値および説明が取り込まれます。つまり、表題として表示する項目を選択できます。例えば、ホスト画面に「**5. 表示**」というアクションが存在する場合、表題表示として「**5**」、「**表示**」、または「**5. 表示**」を選択できます。

サブファイル・アクションを表示 (Show subfile actions)
ポップアップ・ダイアログに加え、レンダリングされたサブファイルとサブファイル・アクションの説明文 (例: **2=Add**) を表示する場合は、このボックスを選択します。このチェック・ボックスは、「**説明を表示**」を選択した場合には使用できません。

選択時に自動的に実行依頼
これを選択した場合、ポップアップ・ダイアログで選択操作が行われると、その選択内容が実行依頼されます。

送信ボタンを表示
これを選択した場合、サブファイルの下に送信ボタンが表示されます。

送信ボタンの表題
オプション。送信ボタンの表題を指定します。

ボタンのスタイル・クラス Web のみ
オプション。生成される送信ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ボタン行のスタイル・クラス Web のみ
オプション。送信ボタンを含む生成されるテーブル行に関連する CSS スタイル・クラスです。デフォルト値は **HATSTABLEHEADER** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ランチャー・タイプ
ポップアップ・ダイアログを開くときに「**イメージ**」、「**ボタン**」、または「**リンク**」のいずれを使用するかを指定します。

イメージ・ファイル名 (Image filename)
オプション。ランチャーで使用するイメージのファイル名。

表題 オプション。ランチャー・ボタンまたはリンクで使用する表題。

リンクのスタイル・クラス Web のみ
オプション。生成されたアクション・リンクに関連する CSS スタイル・クラスです。HTML リンク・タグの class 属性の値には、こ

の値がセットされます。デフォルト値は **HATSLINK** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ボタンのスタイル・クラス **Web のみ**

オプション。生成されたアクション・ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

リンクの表題を閉じる **Web のみ**

「クローズ・リンク」アクションで表示される表題。

1 レコードあたりの行数 (Rows per record)

行の代替色の表示 **RCP のみ**

行を代替色でレンダリングするには、このボックスを選択します。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

行数 (Number of rows)

サブファイル内の 1 レコードあたりの固定行数を指定します。

列の配置 **Web のみ**

このウィジェットは、データの基本列を表示し、必要に応じて追加詳細列を表示する詳細セクションを持つように構成できます。この設定は、モバイル装置にテーブル・データを表示するときに便利です。これらの設定および HATS Toolkit GUI に表示されないこのオプションの他の設定について詳しくは、54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

基本列 このフィールドを使用して、表示する列およびその順序を指定します。フィールドでは、拡張簡略記述を使用して、列番号の数値での任意の順序でリスト表示をすることができ、列を複数回表示したり、またはまったく表示させないこともできます (例: 「**1,3,2,5-7,2-3,9**」)。

2 つの数字の間のハイフン (-) は列の範囲を示します。アスタリスク文字 (*) を使用して、使用可能な最後の列を示すことができます。アスタリスクのみは、列 1 から n (n は使用可能な最後の列) を意味します。また、5-* の省略形として 5* を使用することもできます。1-* と * は同じ意味です。

Web プロジェクトでは、この設定のデフォルトは *****、つまりすべての列になります。モバイル装置に最適化された Web プロジェクトでは、この設定のデフォルトは **1-2**、つまり列 1 と列 2 になっています。

注: 数値はコンポーネントによってウィジェットに提供された列を表します。ウィジェットに提供される前に、除外されるかどうかわからないコンポーネントによって最初に認識された列を表すものではありません。

詳細列 このオプションのフィールドを使用して、要求されたときに表示する詳細列およびその順序を指定します。フィールドでは、拡張簡略記述を使用して、数値での任意の順序でリスト表示することができます。

Web プロジェクトでは、この設定のデフォルトは空、つまり詳細列セクションはありません。モバイル装置に最適化された Web プロジェクトでは、この設定のデフォルトは **3***、つまり列 3 から最後の列になっています。

詳細列をサーバーに保持

このオプションを選択して、ユーザーによって要求されるまで詳細列セクションをサーバーに保持します。これによって、初期レンダリングでの HTML 出力の量を削減できます。デフォルトはクリアされます。

強調表示された行 RCP のみ

レンダリング時に、強調表示する行の集合を、コンマで区切って指定するか、または範囲指定します。この設定は、生成されたテーブル内の重要な行を強調表示する場合に便利です。この値で表される行は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

強調表示された列 RCP のみ

レンダリング時に強調表示する列の集合を、コンマで区切って指定するか、または範囲指定します。この設定は、生成されたテーブル内の重要な列を強調表示する場合に便利です。この値で表される列は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

格子線の表示 RCP のみ

レンダリングされるサブファイルで格子線を表示するには、このボックスを選択します。

列見出しのソース RCP のみ

列ヘッダーのテキストを、サブファイル・ヘッダー・コンポーネントから取り込むか (「**コンポーネントから**」)、またはユーザーが指定するか (「**カスタム**」) を、ドロップダウンから選択します。サブファイル・ヘッダー・コ

ンポーネントから取り込む場合は、詳細について 248 ページの『ヘッダー』のコンポーネントの説明を参照してください。

列見出し (コンマ区切り) **RCP のみ**

「**列見出しのソース**」を「**カスタム**」に設定した場合に、複数の列ヘッダーをコンマで区切ってここに入力します。

クラシック見出しスタイルを使用 **RCP のみ**

クラシック見出しスタイルを表示させるには、このボックスを選択します。また、「**列見出しのソース**」が「**コンポーネントから**」に設定されており、サブファイル・ヘッダー・コンポーネントにより複数行ヘッダーが検出されると、複数行ヘッダーが表示されます。デフォルトの SWT テーブル制御ヘッダーを単一行でのみ表示するには、このボックスをクリアします。

ヘッダーのスペースをトリム (Trim spaces on headers) **RCP のみ**

これを選択した場合、ヘッダー行のセルの内部にあるテキストの両端から空白 (余分なスペース) が除去されます。

前景色を使用可能にする

これを選択した場合、ホスト画面の前景色がレンダリングされます。

Web アプリケーションの場合は、前景色はその色を表す CSS スタイルシート・クラスにマップされます。例えば、ホスト画面フィールドが **RED** とマークされている場合、フィールド・ウィジェットは、そのフィールド用に生成された HTML を、クラス名属性が **HRED** に設定されているタグで囲みます。この処理によって、ユーザーは、生成される Web ページ上にホスト画面の色を再マップすることができます。

注: blacktheme.css は、ホスト画面フィールドの前景色を直接マップします (例えば、ホスト画面上の赤色のテキストは、生成された Web ページ上で赤色のテキストとして表示されます)。ただし、monochrome.css などの他のスタイルシート・ファイルは、フィールドの色を別の方法でマップし、一貫性のある、最新のスタイルを作成しようとします。

リッチ・クライアント・アプリケーションの場合、色はリッチ・クライアント・テンプレートによってマップされます。

注: この設定は、「**色のオーバーライド**」>「**前景色**」の設定によってオーバーライドされます。

フィールドのスタイル・クラス **Web のみ**

オプション。生成されたフィールドに関連する CSS スタイル・クラスです。HTML タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSFIELD** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

拡張属性を使用可能にする

これを選択した場合、拡張フィールド属性 (明滅、反転表示、下線、および列分離子) が指定されたスタイルで表示されます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

明滅スタイル **Web のみ**

ホスト画面における明滅テキストのレンダリングに使用する CSS スタイル設定。

反転表示スタイル (Reverse video style) **Web のみ**
ホスト画面における反転表示テキストのレンダリングに使用する CSS スタイル設定。

下線スタイル (Underline style) **Web のみ**
ホスト画面における下線付きテキストのレンダリングに使用する CSS スタイル設定。

列分離子スタイル (Column separator style) **Web のみ**
ホスト画面における列分離子のレンダリングに使用する CSS スタイル設定。

入力フィールドの下線文字を除去 (Strip underscores on input field)
テキストをレンダリングするときにテキストから下線文字を除去する場合は、このボックスを選択します。

入力フィールドのスペースをトリム
入力フィールドから先行および後続のスペースをトリムするには、このボックスを選択します。

入力フィールドのカーソル位置オプションを使用可能にする **Web のみ**
このボックスを選択して、ユーザーが入力フィールドでデータ入力モードからカーソル位置決めモードに切り替えができるようにします。データ入力モードのときは、ユーザーは入力フィールドにデータを入力できます。カーソル位置決めモードのときは、タブ移動または別の方法で、入力フィールドの任意の文字上にカーソルを配置できます。この設定は他のカーソル位置決め機能を持たないモバイル装置用に最適化されている Web プロジェクトでは便利です。HATS Toolkit GUI に表示されないこのオプションの他の設定のついては、54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

スプレッドシート・ファイルの提供
これを選択した場合、テーブル・データをスプレッドシート・ファイル・フォーマットで取得するためのランチャーが表示されます。

ファイル名の接頭部
スプレッドシート・ファイル名の接頭部を指定します。デフォルトは **spreadsheet** です。完全なファイル名の形式は `FileNamePrefix_TimeStamp` です。タイム・スタンプの形式は MMDDYY_HHMMSS (例: spreadsheet_041506_113020) です。

ファイル名の拡張子
スプレッドシート・フォーマット・タイプの拡張子名。オプションは **.csv** および **.xls** です。.csv ファイルは、データ列がコンマによって区切られている完全なテキスト・ファイルです。.xls ファイルは、Microsoft Excel Biff3 ファイルです。デフォルトは **.csv** です。

ランチャー・タイプ
ユーザーに対して表示されるランチャーのタイプ。オプションは、「**リンク**」、「**イメージ**」、および「**ボタン**」です。デフォルトは、「**リンク**」(Web プロジェクト) または「**ボタン**」(リッチ・クライアント プロジェクト) です。

ランチャーの表題
ユーザーに対して表示されるランチャーの表題。デフォルトは、「**ダウンロード**」(Web プロジェクト) または「**エクスポート**」(リッチ・クライアント・プロジェクト) です。

ボタンのスタイル・クラス Web のみ
生成されるランチャー・ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

リンクのスタイル・クラス Web のみ
オプション。生成されるランチャー・リンクに関連する CSS スタイル・クラスです。HTML リンク・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSLINK** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

入力フィールドのスタイル・クラス Web のみ
オプション。生成される入力フィールドに関連する CSS スタイル・クラスです。HTML 入力タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSINPUT** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

ウィンドウのスタイル・クラス Web のみ
オプション。生成されるウィンドウに関連する CSS スタイル・クラスです。デフォルト値は、**HATSPOPUP** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

クローズ・リンクのスタイル・クラス Web のみ
オプション。ウィンドウのクローズに使用するリンクに関連する CSS スタイル・クラスです。デフォルト値は **HATSLINK** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

項目リンクのスタイル・クラス Web のみ
オプション。ウィジェットからの項目の選択で使用するリンクに関連する CSS スタイル・クラスです。デフォルト値は **HATSPOPUPITEMLINK** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス Web のみ
オプション。生成されるテーブルに関連する CSS スタイル・クラスです。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブル・セルのスタイル・クラス Web のみ
オプション。生成されるテーブル内の各セルに関連する CSS スタイル・クラスです。HTML テーブル・セル (td) タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLECELL** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

見出し行のスタイル・クラス **Web のみ**

オプション。生成されるテーブル内の各ヘッダー行または列に関連する CSS スタイル・クラスです。HTML テーブルの行またはセル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLEHEADER** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

奇数行のスタイル・クラス **Web のみ**

オプション。テーブル内の奇数番の各行に関連する CSS スタイル・クラスです。奇数行番号の各 HTML テーブル行タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLEODDROW** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

偶数行のスタイル・クラス **Web のみ**

オプション。テーブル内の偶数番の各行に関連する CSS スタイル・クラスです。偶数行番号の各 HTML テーブル行タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLEEVENROW** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**

オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド **RCP のみ**

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド **RCP のみ**

前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

注: リッチ・クライアント・プロジェクトでは、このウィジェットの各種色設定が次の順序 (最も高い優先度から低い優先度への順序) で適用されます。

1. 強調表示された行/列
2. 色のオーバーライド
3. 拡張属性を使用可能にする/前景色を使用可能にする
4. 行の代替色の表示

fieldTextAlignment **Web のみ**

Web プロジェクトで Sans Serif などの非モノスペース・フォントを使用すると、レンダリングされたデータはワード間のスペーシングの問題で適切には位置合わせされません。ソース設定 fieldTextAlignment を使用して、データの位置合わせを修正できます。この設定をプロジェクト全体で有効にする

には、application.hap ファイルのソース・ビューを開き、SubfilePopupWidget ウィジェットのクラスを見つけ、次の例に示すように、設定値を NO_SEGMENTING に変更します。

```
<class name="com.ibm.hats.transform.widgets.SubfilePopupWidget">
.
   <setting name="fieldTextAlignment" value="NO_SEGMENTING"/>
.
</class>
```

この設定を変換内のコンポーネント・レベルで有効にするには、変換のソース・ビューを開き、SubfilePopupWidget ウィジェットの HATS:Component タグを見つけ、次の例に示すように、設定値を NO_SEGMENTING に変更します。

```
...
widget="com.ibm.hats.transform.widgets.SubfilePopupWidget"
...
widgetSettings="...|fieldTextAlignment:NO_SEGMENTING|..."
...
```

この設定の値は、WORD_BY_WORD および NO_SEGMENTING です。デフォルト値は **WORD_BY_WORD** です。

注: サブファイル・コンポーネントでデータ認識に「テーブル (可視)」認識オプションを使用している場合、この設定は効果がありません。

テーブル

テーブル・ウィジェットは、GUI テーブルをレンダリングします。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- テーブル
- テーブル (フィールド)
- テーブル (可視)

注: このウィジェットを使用する際の DBCS に関する考慮事項については、548 ページの『SBCS での最大長の除去』を参照してください。

次の図に、変換においてテーブル (フィールド)・コンポーネントの例のデータを入力として使用するテーブル・ウィジェットの表示例を示します。

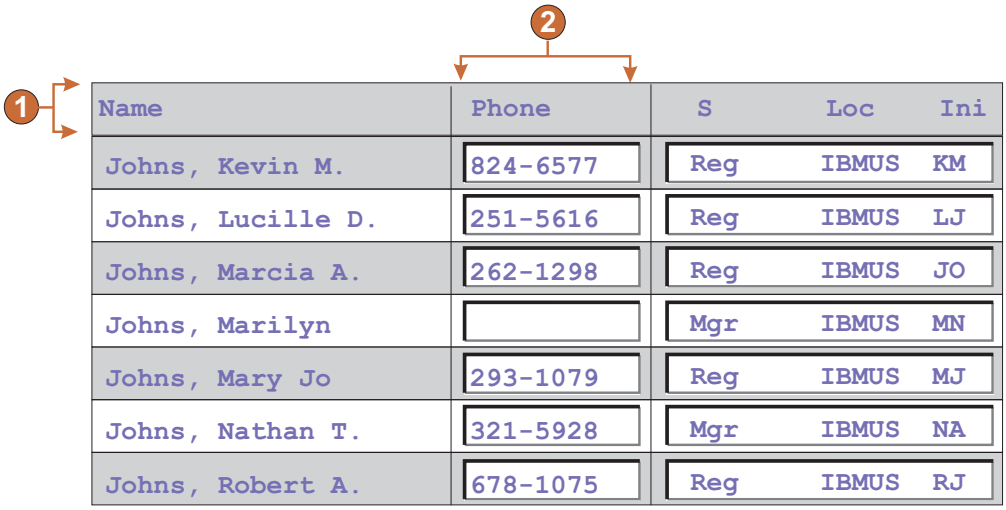


図 56. テーブル・ウィジェットの例

1. ヘッダー行 (1)
2. ヘッダー列 (指定しない)

以下の設定が、このウィジェットに対して構成できます。

ヘッダー行 (Header row) **Web のみ**

テーブル・ヘッダーとしてレンダリングされるコンマで区切られた、または範囲指定による行の集合です。この設定は、生成されたテーブル内の重要な行を強調表示する場合に便利です。この値で表される行は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。「**見出し行のスタイル・クラス**」設定の説明を参照してください。

強調表示された行 **RCP のみ**

レンダリング時に、強調表示する行の集合を、コンマで区切って指定するか、または範囲指定します。この設定は、生成されたテーブル内の重要な行を強調表示する場合に便利です。この値で表される行は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

強調表示された列 **RCP のみ**

レンダリング時に強調表示する列の集合を、コンマで区切って指定するか、または範囲指定します。この設定は、生成されたテーブル内の重要な列を強調表示する場合に便利です。この値で表される列は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター

(http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

行の代替色の表示 **RCP のみ**

行を代替色でレンダリングするには、このボックスを選択します。RGB カラー値は、テンプレートから返される ITableColorProvider オブジェクトによって決まります。RcpTemplate クラスおよび ITableColorProvider クラスについて詳しくは、HATS インフォメーション・センター (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API リファレンス』セクションを参照してください。

格子線の表示 **RCP のみ**

レンダリングされるテーブルで格子線を表示するには、このボックスを選択します。

列見出しのソース **RCP のみ**

列ヘッダーのテキストを、テーブル・コンポーネントから取り込むか (「**コンポーネントから**」)、またはユーザーが指定するか (「**カスタム**」) を、ドロップダウンから選択します。テーブル・コンポーネントから取り込む場合は、詳細について 252 ページの『テーブル』の「**タイトル行の数**」設定のコンポーネント説明を参照してください。

列見出し (コンマ区切り) **RCP のみ**

「**列見出しのソース**」を「**カスタム**」に設定した場合に、複数の列ヘッダーをコンマで区切ってここに入力します。

ヘッダーのスペースをトリム (Trim spaces on headers)

これを選択した場合、ヘッダー行のセルの内部にあるテキストの両端から空白 (余分なスペース) が除去されます。

ヘッダー列 **Web のみ**

ヘッダーとしてレンダリングされるコンマで区切られた、または範囲指定によるテーブル列の集合です。この設定は、生成されたテーブル内の重要な列 (通常は最初の列) を強調表示する場合に便利です。この値で表される列は、実際のホスト画面ではなく、認識されるテーブルに基づきます。値の例: **1,2,3....**、**1,2-4**、**3-5**。「**見出し行のスタイル・クラス**」設定の説明を参照してください。

入力を使用不可にする

これを選択した場合、レンダリングされた入力フィールドは読み取り専用になります。読み取り専用の入力フィールドは通常の入力フィールドのように表示されますが、ユーザーはこのフィールドの内容を変更できません。これは、無保護フィールドの内容をユーザーに対して表示するが、ユーザーがフィールドの内容を変更できないようにする場合に便利です。

列の配置 **Web のみ**

このウィジェットは、データの基本列を表示し、必要に応じて追加詳細列を表示する詳細セクションを持つように構成できます。この設定は、モバイル装置にテーブル・データを表示するときに便利です。これらの設定および

HATS Toolkit GUI に表示されないこのオプションの他の設定について詳しくは、54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

基本列 このフィールドを使用して、表示する列およびその順序を指定します。フィールドでは、拡張簡略記述を使用して、列番号の数値での任意の順序でリスト表示をすることができ、列を複数回表示したり、またはまったく表示させないこともできます (例: 「**1,3,2,5-7,2-3,9**」)。

2 つの数字の間のハイフン (-) は列の範囲を示します。アスタリスク文字 (*) を使用して、使用可能な最後の列を示すことができます。アスタリスクのみは、列 1 から n (n は使用可能な最後の列) を意味します。また、5-* の省略形として 5* を使用することもできます。1-* と * は同じ意味です。

Web プロジェクトでは、この設定のデフォルトは *****、つまりすべての列になります。モバイル装置に最適化された Web プロジェクトでは、この設定のデフォルトは **1-2**、つまり列 1 と列 2 になっています。

注: 数値はコンポーネントによってウィジェットに提供された列を表します。ウィジェットに提供される前に、除外されるかどうかわからないコンポーネントによって最初に認識された列を表すものではありません。

詳細列 このオプションのフィールドを使用して、要求されたときに表示する詳細列およびその順序を指定します。フィールドでは、拡張簡略記述を使用して、数値での任意の順序でリスト表示することができます。

Web プロジェクトでは、この設定のデフォルトは空、つまり詳細列セクションはありません。モバイル装置に最適化された Web プロジェクトでは、この設定のデフォルトは **3***、つまり列 3 から最後の列になっています。

詳細列をサーバーに保持
このオプションを選択して、ユーザーによって要求されるまで詳細列セクションをサーバーに保持します。これによって、初期レンダリングでの HTML 出力の量を削減できます。デフォルトはクリアされます。

入力フィールドの下線文字を除去 (Strip underscores on input field)
テキストをレンダリングするときにテキストから下線文字を除去する場合は、このボックスを選択します。

入力フィールドのスペースをトリム
入力フィールドから先行および後続のスペースをトリムするには、このボックスを選択します。

入力フィールドのカーソル位置オプションを使用可能にする **Web のみ**
このボックスを選択して、ユーザーが入力フィールドでデータ入力モードからカーソル位置決めモードに切り替えができるようにします。データ入力モードのときは、ユーザーは入力フィールドにデータを入力できます。カーソ

ル位置決めモードのときは、タブ移動または別の方法で、入力フィールドの任意の文字上にカーソルを配置できます。この設定は他のカーソル位置決め機能を持たないモバイル装置用に最適化されている Web プロジェクトでは便利です。HATS Toolkit GUI に表示されないこのオプションの他の設定のついては、 54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

前景色を使用可能にする

これを選択した場合、ホスト画面の前景色がレンダリングされます。

Web アプリケーションの場合は、前景色はその色を表す CSS スタイルシート・クラスにマップされます。例えば、ホスト画面フィールドが **RED** とマークされている場合、フィールド・ウィジェットは、そのフィールド用に生成された HTML を、クラス名属性が **HRED** に設定されているタグで囲みます。この処理によって、ユーザーは、生成される Web ページ上にホスト画面の色を再マップすることができます。

注: blacktheme.css は、ホスト画面フィールドの前景色を直接マップします (例えば、ホスト画面上の赤色のテキストは、生成された Web ページ上で赤色のテキストとして表示されます)。ただし、monochrome.css などの他のスタイルシート・ファイルは、フィールドの色を別の方法でマップし、一貫性のある、最新のスタイルを作成しようとします。

リッチ・クライアント・アプリケーションの場合、色はリッチ・クライアント・テンプレートによってマップされます。

注: この設定は、「**色のオーバーライド**」>「**前景色**」の設定によってオーバーライドされます。

拡張属性を使用可能にする

これを選択した場合、拡張フィールド属性 (明滅、反転表示、下線、および列分離子) が表示されます。また、3270 Web アプリケーションの場合、拡張フィールドの色もマップされます (詳細については、前の設定を参照してください)。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

リッチ・クライアント・アプリケーションの場合、拡張フィールド属性はリッチ・クライアント・テンプレートによってマップされます。

明滅スタイル **Web のみ**

ホスト画面における明滅テキストのレンダリングに使用する CSS スタイル設定。

反転表示スタイル (Reverse video style) **Web のみ**

ホスト画面における反転表示テキストのレンダリングに使用する CSS スタイル設定。

下線スタイル (Underline style) **Web のみ**

ホスト画面における下線付きテキストのレンダリングに使用する CSS スタイル設定。

列分離子スタイル (Column separator style) **Web のみ**

ホスト画面における列分離子のレンダリングに使用する CSS スタイル設定。

スプレッドシート・ファイルの提供
これを選択した場合、テーブル・データをスプレッドシート・ファイル・フォーマットで取得するためのランチャーが表示されます。

ファイル名の接頭部
スプレッドシート・ファイル名の接頭部を指定します。デフォルトは **spreadsheet** です。完全なファイル名の形式は `FileNamePrefix_TimeStamp` です。タイム・スタンプの形式は MMDDYY_HHMMSS (例: spreadsheet_041506_113020) です。

ファイル名の拡張子
スプレッドシート・フォーマット・タイプの拡張子名。オプションは **.csv** および **.xls** です。.csv ファイルは、データ列がコンマによって区切られている完全なテキスト・ファイルです。.xls ファイルは、Microsoft Excel Biff3 ファイルです。デフォルトは **.csv** です。

ランチャー・タイプ
ユーザーに対して表示されるランチャーのタイプ。オプションは、「**リンク**」、「**イメージ**」、および「**ボタン**」です。デフォルトは、「**リンク**」(Web プロジェクト) または「**ボタン**」(リッチ・クライアント プロジェクト) です。

ランチャーの表題
ユーザーに対して表示されるランチャーの表題。デフォルトは、「**ダウンロード**」(Web プロジェクト) または「**エクスポート**」(リッチ・クライアント・プロジェクト) です。

ボタンのスタイル・クラス **Web のみ**
生成されるランチャー・ボタンに関連する CSS スタイル・クラスです。HTML ボタン・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSBUTTON** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

リンクのスタイル・クラス **Web のみ**
オプション。生成されるランチャー・リンクに関連する CSS スタイル・クラスです。HTML リンク・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSLINK** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス **Web のみ**
オプション。生成されるテーブルに関連する CSS スタイル・クラスです。HTML テーブル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

テーブル・セルのスタイル・クラス **Web のみ**
オプション。生成されるテーブル内の各セルに関連する CSS スタイル・クラスです。HTML テーブル・セル (td) タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLECELL** です。詳細については、374 ページの『スタイル・シートの使用』を参照してください。

見出し行のスタイル・クラス **Web のみ**

オプション。生成されるテーブル内の各ヘッダー行または列に関連する CSS スタイル・クラスです。HTML テーブルの行またはセル・タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSTABLEHEADER** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

奇数行のスタイル・クラス **Web のみ**

オプション。テーブル内の奇数番の各行に関連する CSS スタイル・クラスです。奇数番号行の各 HTML テーブル行タグのクラス属性値には、この値がセットされます。デフォルト値は **HATSTABLEODDROW** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

偶数行のスタイル・クラス **Web のみ**

オプション。テーブル内の偶数番の各行に関連する CSS スタイル・クラスです。偶数番号行の各 HTML テーブル行タグのクラス属性値には、この値がセットされます。デフォルト値は **HATSTABLEEVENROW** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**

オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド **RCP のみ**

デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド **RCP のみ**

前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

注: リッチ・クライアント・プロジェクトでは、このウィジェットの各種色設定が次の順序 (最も高い優先度から低い優先度への順序) で適用されます。

1. 強調表示された行/列
2. 色のオーバーライド
3. 拡張属性を使用可能にする/前景色を使用可能にする
4. 行の代替色の表示

テキスト入力

テキスト入力ウィジェットは、HTML 入力フィールドのレンダリングを担当します。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- コマンド行
- 入力フィールド

注: このウィジェットを使用する際の DBCS に関する考慮事項については、 548 ページの『SBCS での最大長の除去』を参照してください。

次の図に、変換においてコマンド行コンポーネントの例のデータを入力として使用するテキスト入力ウィジェットの表示例を示します。

図 57. テキスト入力ウィジェットの例

1. 表題 (コンポーネントから)

以下の設定が、このウィジェットに対して構成できます。

サイズをオーバーライド
これを選択した場合、指定したサイズ値が生成された各入力フィールドに設定されます。この設定は、大きなホスト入力フィールドのサイズを視覚的に変更する場合に便利です。クリアした場合、入力フィールドのサイズは認識する側のコンポーネントによって決定されます。

サイズ 各入力フィールドのサイズ。

注: この値はブラウザーによって解釈されます (入力フィールドが、別のブラウザー上でどのようにレンダリングされるかは、保証されません)。

最大長をオーバーライド
これを選択した場合、各生成入力フィールドに指定された最大長の値が設定されます。この設定は、入力フィールドに入力可能な文字数を手動で制限する場合に便利です。クリアした場合は、入力文字の最大許容数は、認識する側のコンポーネントによって決定されます。

最大長 各入力フィールドに入力可能な最大文字数です。

表題ソース (Caption source)
生成される入力フィールドの表題を決定する方法を指定します。オプション: **「コンポーネントから」** (コンポーネントから取り出した表題を使用) または**「カスタム」**(指定された表題を使用)。

カスタム表題
オプション。入力フィールドの表題を指定します。

注: チェック・ボックスの表題を生成しない場合は、この設定をブランクのままにしてください。

1 行あたりの列数
次の行に折り返されるまでに、水平方向に表示される入力フィールドの数です。

注:

1. この設定はデフォルト・レンダリングでは適用されません。
2. ウィジェットのプレビューは必ずしも全ページのプレビューと一致するとは限りません。これはウィジェットのプレビューが単にコンポーネントの出力をレンダリングしているだけであるために起こります。ウィジェットのプレビューは、全ページのプレビューで行われているように、レンダリングしたときに画面の配置を保存しようとはしません。

読み取り専用

これを選択した場合、レンダリングされた入力フィールドは読み取り専用になります。読み取り専用の入力フィールドは通常の入力フィールドのように表示されますが、ユーザーはこのフィールドの内容を変更できません。これは、無保護フィールドの内容をユーザーに対して表示するが、ユーザーがフィールドの内容を変更できないようにする場合に便利です。

入力フィールドの下線文字を除去 (Strip underscores on input field)

テキストをレンダリングするときにテキストから下線文字を除去する場合は、このボックスを選択します。

入力フィールドのスペースをトリム

入力フィールドから先行および後続のスペースをトリムする場合は、このボックスを選択します。

入力フィールドのカーソル位置オプションを使用可能にする Web のみ

このボックスを選択して、ユーザーが入力フィールドでデータ入力モードからカーソル位置決めモードに切り替えができるようにします。データ入力モードのときは、ユーザーは入力フィールドにデータを入力できます。カーソル位置決めモードのときは、タブ移動または別の方法で、入力フィールドの任意の文字上にカーソルを配置できます。この設定は他のカーソル位置決め機能を持たないモバイル装置用に最適化されている Web プロジェクトでは便利です。HATS Toolkit GUI に表示されないこのオプションの他の設定のついては、54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

前景色を使用可能にする

これを選択した場合、ホスト画面の前景色がレンダリングされます。

Web アプリケーションの場合は、前景色はその色を表す CSS スタイルシート・クラスにマップされます。例えば、ホスト画面フィールドが **RED** とマークされている場合、フィールド・ウィジェットは、そのフィールド用に生成された HTML を、クラス名属性が **HRED** に設定されているタグで囲みます。この処理によって、ユーザーは、生成される Web ページ上にホスト画面の色を再マップすることができます。

注: blacktheme.css は、ホスト画面フィールドの前景色を直接マップします (例えば、ホスト画面上の赤色のテキストは、生成された Web ページ上で赤色のテキストとして表示されます)。ただし、monochrome.css などの他のスタイルシート・ファイルは、フィールドの色を別の方法でマップし、一貫性のある、最新のスタイルを作成しようとします。

リッチ・クライアント・アプリケーションの場合、色はリッチ・クライアント・テンプレートによってマップされます。

注: この設定は、「**色のオーバーライド**」>「**前景色**」の設定によってオーバーライドされます。

3270 数値ロックを有効にする
選択されている場合、ユーザーは、0 から 9 までの数字、正符号、負符号、ピリオド、コンマ、およびヒンディ数字のみを、3270 数値フィールドに入力できます。クリアされている場合、任意の文字を入力できます。デフォルトではクリアされています。

注: Web アプリケーションの場合、必要に応じて lxgwfunctions.js ファイルの allowNumLockOnly() 関数および allowNumLockOnlyForIEMobile() 関数で文字コード値を使用して、受け入れられるキーのリストをカスタマイズできます。

拡張属性を使用可能にする
これを選択した場合、拡張フィールド属性 (明滅、反転表示、下線、および列分離子) が表示されます。また、3270 Web アプリケーションの場合、拡張フィールドの色もマップされます (詳細については、前の設定を参照してください)。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

リッチ・クライアント・アプリケーションの場合、拡張フィールド属性はリッチ・クライアント・テンプレートによってマップされます。

明滅スタイル **Web のみ**
ホスト画面における明滅テキストのレンダリングに使用する CSS スタイル設定。

反転表示スタイル (Reverse video style) **Web のみ**
ホスト画面における反転表示テキストのレンダリングに使用する CSS スタイル設定。

下線スタイル (Underline style) **Web のみ**
ホスト画面における下線付きテキストのレンダリングに使用する CSS スタイル設定。

列分離子スタイル (Column separator style) **Web のみ**
ホスト画面における列分離子のレンダリングに使用する CSS スタイル設定。

入力フィールドのスタイル・クラス **Web のみ**
オプション。生成される入力フィールドに関連する CSS スタイル・クラスです。HTML 入力タグの class 属性の値には、この値がセットされます。デフォルト値は **HATSINPUT** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

表題のスタイル・クラス **Web のみ**
オプション。生成される入力フィールドの表題に関連する CSS スタイル・クラスです。デフォルト値は **HATSCAPTION** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

テーブルのスタイル・クラス **Web のみ**
オプション。生成されるテーブルに関連する CSS スタイル・クラスです。入力フィールドが複数レンダリングされる場合、1 つの HTML テーブルが生成され、それらのフィールドを囲みます。HTML テーブル・タグの class

属性の値には、この値がセットされます。デフォルト値は **HATSTABLE** です。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

スタイル **Web のみ**
オプション。スタイル・プロパティー・ダイアログを表示するには、このフィールドの横にあるランチャー・ボタンを使用します。このダイアログでは、CSS の知識がなくてもウィジェットのフォント、色、またはその他のスタイル設定を変更することができます。詳細については、 374 ページの『スタイル・シートの使用』を参照してください。

フォントのオーバーライド **RCP のみ**
デフォルトのフォント名、フォント・スタイル、フォント・サイズを指定変更する場合は、このオプションを選択して、提供されている「**フォント**」設定コントロールを使用します。

色のオーバーライド **RCP のみ**
前景色と背景色を指定変更する場合は、このオプションを選択して、提供されている「**前景色**」と「**背景色**」の設定コントロールを使用します。

ツールバー RCP のみ

ツールバー・ウィジェットにより、ファンクション・キーと選択リスト項目をツールバー上のボタンとしてレンダリングできます。これらの項目は、「変換」ビューのメイン・ツールバーまたはスタンドアロン・ツールバーのいずれかにレンダリングできます。これは、パネル上にセクションを作成する場合、および特定のセクションにツールバーを提供する場合に便利です。「変換」ビューのツールバーに項目をレンダリングする場合は、ボタンのスタイル (テキスト、イメージ、またはテキストとイメージの両方) はプロジェクト・レベルのツールバー設定によって決まります。プロジェクト・レベルのツールバー設定の詳細については、 121 ページの『ツールバー RCP のみ』を参照してください。スタンドアロン・ツールバーに項目をレンダリングする場合は、このウィジェットの設定を使用してツールバーのスタイルを制御します。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- ファンクション・キー
- 選択リスト

次の図に、ファンクション・キー・コンポーネントのデータを入力として使用してツールバー・ウィジェットが「変換」ビュー・ツールバーで表示される例を示します。

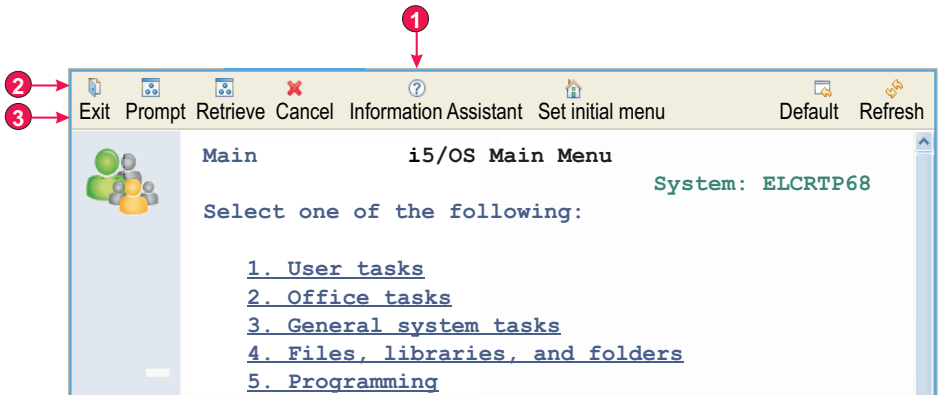


図 58. ツールバー・ウィジェットの例

1. メイン・ツールバーへの項目の提供
2. ツールバー項目の表示形式 (テキストおよびイメージ)

   注: この設定 (**ツールバー項目の表示形式**) は、プロジェクト設定エディターの「**レンダリング**」タブの「**ツールバー**」メイン設定で構成されます。この値は、ツールバー・ウィジェットに対して「**メイン・ツールバーへの項目の提供**」が選択されている場合に適用されます。
3. 表題のタイプ (説明を表示)

以下の設定が、このウィジェットに対して構成できます。

メイン・ツールバーへの項目の提供

「変換」ビューのメイン・ツールバーに項目をレンダリングするには、このボックスを選択します。これを選択した場合、プロジェクト設定エディターの「**レンダリング**」タブのプロジェクト・レベルの「**ツールバー**」設定に基づいてツールバー項目がレンダリングされます。例えば、このウィジェットに対して「**単一のツールバー項目の下に項目をレンダリング**」設定が選択されている場合は、プロジェクト・レベルの「**ツールバー項目の表示形式**」設定により許可されている場合にのみ、その「**表題**」および「**イメージ**」設定が適用されます。詳しくは、 121 ページの『ツールバー RCP のみ』を参照してください。

スタンドアロン・ツールバーの項目をレンダリングするには、このボックスをクリアします。このオプションを使用する場合は、次に説明する「**項目の表示方法**」設定を使用して項目のレンダリング方法を制御します。

項目の表示方法

スタンドアロン・ツールバーの項目をレンダリングする場合は、この設定を使用して各項目の表示方法を制御します。各項目の表示オプションは、「**テキスト**」、「**イメージ**」、または「**テキストとイメージの両方**」です。

単一のツールバー項目の下に項目をレンダリング

このウィジェットによりレンダリングされるすべての項目を、ツールバー上の 1 つの項目の下のメニュー項目として表示することを指定するには、このボックスを選択します。

表題 単一ツールバー項目に使用するテキスト表題を指定します。「**メイン・ツールバーへの項目の提供**」設定によっては、プロジェクト・レベルの「**ツールバー項目の表示形式**」設定またはウィジェット・レベルの「**項目の表示形式**」設定が「**イメージ**」に設定されている場合に、この設定が無効になることがあります。詳しくは、上記の「**メイン・ツールバーへの項目の提供**」設定の説明を参照してください。デフォルトは 「**アクション**」です。

イメージ
単一ツールバー項目に使用するイメージを指定します。ドロップダウンからイメージを選択するか、または「**インポート**」ボタンを使用してイメージをインポートします。「**メイン・ツールバーへの項目の提供**」設定によっては、プロジェクト・レベルの「**ツールバー項目の表示方法**」設定またはウィジェット・レベルの「**項目の表示形式**」設定が「**テキスト**」に設定されている場合に、この設定が無効になることがあります。詳しくは、上記の「**メイン・ツールバーへの項目の提供**」設定の説明を参照してください。

表題のタイプ
この設定を使用して、各項目に表示されるテキスト表題を制御します。「**項目の表示方法**」が「**イメージ**」に設定されている場合は、この設定は無効です。オプションは「**先行トークンを表示**」、「**説明を表示**」、または「**両方を表示**」です。例えば、ホスト・テキストが F3=Exit の場合、「**先行トークンを表示**」を指定すると **F3** が表示され、「**説明を表示**」を指定すると **Exit** が表示され、「**両方を表示**」を指定すると **F3=Exit** が表示されます。

吹き出しヘルプのソース
ツールバーの項目の上に移動するとツールチップとして表示されるテキストの内容を制御します。オプションは「**先行トークンを表示**」、「**説明を表示**」、または「**両方を表示**」です。例えば、ホスト・テキストが F3=Exit の場合、「**先行トークンを表示**」を指定すると **F3** が表示され、「**説明を表示**」を指定すると **Exit** が表示され、「**両方を表示**」を指定すると **F3=Exit** が表示されます。

プロジェクト・レベル・イメージ・マッピングの使用
ツールバーの項目に表示するイメージを、プロジェクト・レベルのイメージ・マッピングで検索することを指定するには、このボックスを選択します。最初にコンポーネント・レベルまたはデフォルト・レンダリング・レベルのマッピングが検索された後で、プロジェクト・レベル・マッピングが検索されます。

プロジェクト・レベルのイメージ・マッピング
ツールバーの項目でイメージをレンダリングする場合は、ファンクション・キーまたは選択リストのソース・テキストを特定のイメージにマップできます。例えば、ファンクション・キーのソース・テキスト F3=Exit を、先行トークン (F3)、説明 (Exit)、または完全な表題 (F3=Exit) のいずれかに基づくイメージにマップできます。他のウィジェット設定とは異なり、この設定ではプロジェクト設定エディターでプロジェクト・レベルのイメージ・マップを構成し、これらのマッピングをコンポーネント・レベルまたはデフォルト・レンダリング・レベルで (単なる上書きではなく) 拡張できます。こ

れにより、汎用のプロジェクト・レベルのマッピングを構成し、コンポーネント・レベルまたはデフォルトのレンダリング・レベルで小規模な変更を行うことができます。

「**追加**」ボタンを使用して、新規マッピングを追加します。次のように設定されている「**条件を追加**」ダイアログが表示されます。

テキスト
ファンクション・キーまたは選択リストのソース・テキストのどの部分で、「**値**」フィールドに指定された値を検索するかを定義します。オプションは「**先行トークン**」、「**説明**」、または「**両方**」です。例えば、ソース・テキストが F3=Exit である場合は、「**値**」フィールドに指定された値と一致するものを検索するときに、「**先行トークン**」 を指定すると F3 が検索され、「**説明**」を指定すると Exit が検索され、「**両方**」を指定すると F3=Exit が検索されます。

演算子 「**値**」フィールドに指定された値を「**テキスト**」フィールドで定義されているテキストで検索するときに、一致として見なされる条件を定義します。オプションは、「**含む**」または「**等しい**」です。「**含む**」では、テキスト内の任意の位置で値が検出された場合に一致とみなされます。この場合、検索では大文字と小文字は区別されません。「**等しい**」では、テキストが値と同等である場合にのみ一致とみなされます。この場合、検索では大文字と小文字が区別されます。

値 「**テキスト**」フィールドで定義されるテキスト内で検索する値を定義します。複数の値を指定するには、縦線 (|) 記号で区切ります (例: Exit|Quit|Leave)。

イメージ
一致が検出された場合に使用するイメージを定義します。ドロップダウン・リストからイメージを選択するか、または「**インポート**」ボタンをクリックしてイメージをインポートします。

プレビュー
選択したイメージのビューを表示します。

イメージ・マッピングをさらに定義する場合は、「**編集**」、「**除去**」、「**上へ**」、および「**下へ**」ボタンを使用します。

注: ツールバー・ウィジェットが使用するイメージを判別する前に、テキスト置換が実行されます。定義するマッピングでは、テキスト置換により項目の先行トークン、説明、またはこの両方が変更されている可能性があることを前提とする必要があります。

defaultImage
イメージが特定のファンクション・キーまたは選択リスト項目にマップされない場合に、この設定を使用して、使用するデフォルト・イメージを指定します。

注: GUI では、この設定は構成可能な設定として示されません。ツールバー・ウィジェット設定のソースで設定する必要があります。例えば、デ

フォルトとして、details.gif イメージをプロジェクトの「リッチ・クライアント・コンテンツ/イメージ」フォルダーで使用する場合、以下のいずれかを行います。

プロジェクト・レベルでデフォルトを指定するには、「HATS プロジェクト」ビューのプロジェクト・フォルダーの「**プロジェクト設定**」ファイルをダブルクリックし、application.hap ファイルのエディターを起動します。「ソース」タブで、<classSettings> タグの中のクラス name="com.ibm.hats.rcp.transform.widgets.SwtToolbarWidget" を見つけ、defaultImage の設定を `<setting name="defaultImage" value="images/details.gif"/>` に変更します。

変換の ComponentRendering コントロールのデフォルト・イメージを設定するには、ComponentRendering オブジェクトの setWidgetSettings メソッドに渡された defaultImage ストリングを更新します。 例えば、`defaultImage:` を `defaultImage:images/details.gif` に変更します。

別のプラグインに含まれているイメージを使用するには、そのプラグイン内のイメージへの絶対パスを指定します。例えば、`defaultImage:` を `defaultImage:org.eclipse.ui/icons/full/obj16/delete_obj.gif` に変更します。

ウィジェット設定 (Dojo)

HATS Dojo ウィジェットの考慮事項と制限

Dojo は、オープン・ソースのテクノロジーであり、継続して開発中です。HATS Dojo ウィジェットを使用する場合の既知の考慮事項および制限 (サポート対象ブラウザーを含む) に関する更新版のリストを確認するには、「Host Access Transformation Services 9.5 - Known issues and workarounds」(http://www.ibm.com/support/docview.wss?uid=swg27046998) を参照してください。 Dojo サポート・ページ (http://dojotoolkit.org/) も参照してください。

以下に、HATS Dojo ウィジェットを使用する際の考慮事項および制限をリストします。

- Dojo ウィジェットは、HATS Web アプリケーション内でのみサポートされます。以下の HATS アプリケーション内ではサポートされません。
  - 双方向
  - モバイル装置
  - リッチ・クライアント
- Dojo ウィジェットは、以下の HATS 実装環境ではサポートされません。
  - 代替レンダリング
  - デフォルト・レンダリング
  - グローバル規則
  - 画面組み合わせ
  - HATS タブ付きフォルダー

- ホスト画面の広い領域を囲む Dojo エレメントは、望ましいレイアウトに収まらないことがあります。単一コンポーネント内の複数のコンポーネント・エレメントによって、レイアウトの問題が生じる場合は、個々のコンポーネント・エレメントをレイアウトするために小さいホスト画面領域を指定することを検討してください。
- HATS Dojo ウィジェットのカスタマイズについては、「*HATS Web アプリケーション・プログラマーズ・ガイド*」の『HATS Dojo ウィジェットのカスタマイズ』を参照してください。

コンボ・ボックス (Dojo) Web のみ

コンボ・ボックス・ウィジェットは、Dojo コンボ・ボックスのレンダリングを担当します。

コンボ・ボックス・ウィジェットの特性は次のとおりです。

- ホストに実行依頼するリスト項目の表題と対応するリスト項目値で構成される値の組のリストが、ドロップダウン・コンボ・ボックスに表示されます。レンダリングされるコンポーネントによっては、値の組をコンポーネントによって指定するか、ウィジェット設定でユーザーが手動で指定することができます。

  注: 表題と値の両方がドロップダウン・リストに表示されます。
- コンボ・ボックス・ウィジェットは編集可能であるため、ユーザーは、リストに表示される値だけでなく、実行時に任意の値を入力できます。

  注: コンボ・ボックス・ウィジェットとフィルター操作選択ウィジェットの主な違いは、コンボ・ボックス・ウィジェットでユーザーは、リストにない値を入力して実行依頼できることです。
- ユーザーがコンボ・ボックスに入力するときに、リストで部分的に一致する値がドロップダウン・リストに表示されます。
- ウィジェットは、多くの API オプションを使用してカスタマイズ可能です。そのようなオプションには、検証および制約オプション、エラー・メッセージ、アイコン、オートコンプリートなどが含まれます。例えば、デフォルトでは、Dojo コンボ・ボックスに対するクライアント・サイドの検証はありません。クライアント・サイドの検証を追加する方法の例を他のウィジェットのカスタマイズとともに確認するには、 「*HATS Web アプリケーション・プログラマーズ・ガイド*」の『HATS Dojo ウィジェットのカスタマイズ』を参照してください。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- コマンド行
- 入力フィールド
- ヒント付き入力フィールド
- 項目の選択
- 選択リスト

次の図に、変換において「**ストリングから設定**」設定を使用して指定したデータを使用するコンボ・ボックス・ウィジェットの表示例を示します。

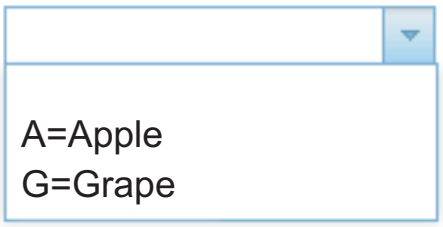


図 59. Dojo コンボ・ボックス・ウィジェットの例

以下の図は、ポップアップ・リストに表示される、部分的に一致した値の例を示しています。この例では、ユーザーは入力テキスト・ボックスに `G` と入力しました。

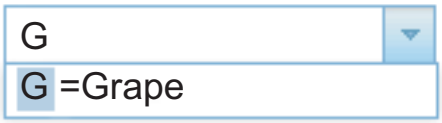


図 60. Dojo コンボ・ボックス・ウィジェットのフィルター操作の例

以下の図に示すように、コンボ・ボックス・ウィジェットでは、フィルター操作選択ウィジェットと異なり、ユーザーは表示されるリストにない値を入力して実行依頼できます。

図 61. Dojo コンボ・ボックス・ウィジェットのユーザー指定入力の例

以下の設定が、このウィジェットに対して構成できます。

グローバル変数から設定
: これを選択した場合、指定したグローバル変数がドロップダウン・リストに設定されます。

  リストの値を含むグローバル変数
  : 値の集合を含む索引付きグローバル変数の名前を指定します。グローバル変数内の各索引の項目がドロップダウン・リストに作成されます。

  共用 HATS のローカル・グローバル変数と共用グローバル変数は同じ名前を持つことができます。共用グローバル変数を使用してリスト項目にデータを取り込む場合は、このボックスを選択します。このボックスをクリアした場合は、ローカル・グローバル変数が使用されます。

リストの表題を含むグローバル変数

オプション。表題の集合を含む索引付きグローバル変数の名前を指定します。この値によって指定されるグローバル変数のサイズは、前の設定で指定されたサイズ以上である必要があります。この索引付きグローバル変数の索引は、グローバル変数の値の索引と一致する必要もあります (その結果、ユーザーに表示される実際の値と表題が同期します)。この値が指定されない場合は、ドロップダウン・リスト内の各項目の表題がその値になります。

ストリングから設定

これを選択した場合、指定したストリングがリストに設定されます。

リスト項目

オプション。ドロップダウン・リストに含まれる項目のストリングを指定します。項目はセミコロン (;) で分離する必要があります。リスト項目の表題とリスト項目の値が異なるようにするには、表題と値を等号 (=) で区切って両方を入力します。例えば、**Apple=A;Grape=G** の値は、2 つの項目「**A=Apple**」と「**G=Grape**」を持つドロップダウン・リストをレンダリングします。最初の項目を選択すると、関連するホスト画面の入力フィールドに「**A**」が挿入されます。

ドロップダウン・リストの項目とホスト画面に挿入される値を同一にするには、項目のみを入力します。例えば、**Apple=A;G** です。この例では、ドロップダウン・リストとホスト画面の入力フィールドに **G** が表示されます。

選択時に自動的に実行依頼

これを選択した場合、ドロップダウン・リストで選択操作が行われると、その選択内容が実行依頼されます。

日付テキスト・ボックス (Dojo) Web のみ

日付テキスト・ボックス・ウィジェットは、ポップアップ・カレンダー日付ピッカーを備えた Dojo 日付テキスト・ボックスのレンダリングを担当します。

このウィジェットは、以下のコンポーネントの提供するデータをレンダリングします。

- 入力フィールド

次の図に、変換における日付テキスト・ボックス・ウィジェットの表示例を示します。

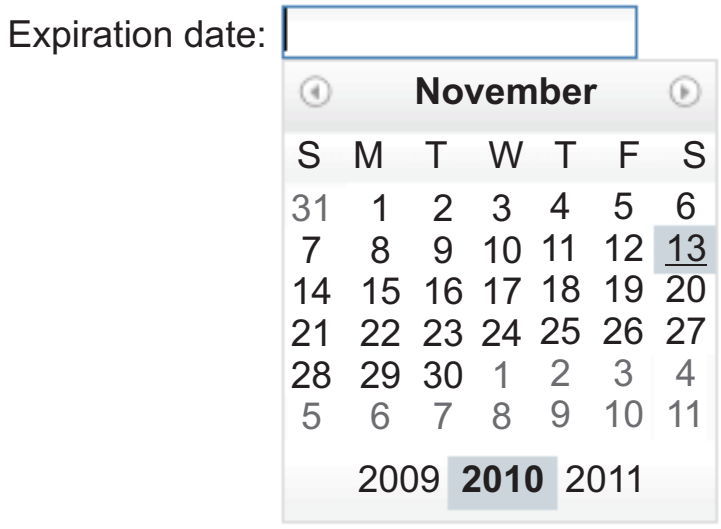


図 62. Dojo 日付テキスト・ボックス・ウィジェットの例

以下の設定が、このウィジェットに対して構成できます。

パターン

必須。ユーザーがポップアップ・カレンダーから日付を選択すると、選択された日付をホスト・アプリケーションの入力フィールドで予期されるフォーマットに正しくマップするようにフォーマット設定する必要があります。この設定は、選択された日付を正しくフォーマット設定するために HATS が使用するパターンを指定します。パターン内の記号の意味については、http://docs.dojocampus.org/dojo/date/locale/format を参照してください。

パターンを入力して保存すると、HATS Toolkit は正しい記号が使用されていることを確認するために以下の変換を実行します。

- 大文字の `D` から小文字の `d`
- 大文字の `Y` から小文字の `y`
- 小文字の `m` から大文字の `M`
- 小文字の `e` から大文字の `E`

選択可能な最も古い日付を制限

これを選択した場合、日付フィールドで指定される値は、ユーザーがポップアップ・カレンダーから選択可能な最も古い日付となります。この設定では、関連する入力フィールドにユーザーが古い日付を手動で直接入力する操作は防止されません。

注: ユーザーがこの範囲外の日付を入力すると、警告シンボルが表示されます。範囲メッセージ (デフォルトまたはユーザー指定) を定義すると、そのメッセージが表示されます。

日付 すべてのロケールで、日付は MM/DD/YYYY 形式で入力します。

選択可能な最も新しい日付を制限

これを選択した場合、日付フィールドで指定される値は、ユーザーがポップ

アップ・カレンダーから選択可能な最新の日付となります。この設定では、関連する入力フィールドにユーザーが新しい日付を手動で直接入力する操作は防止されません。

注: ユーザーがこの範囲外の日付を入力すると、警告シンボルが表示されます。範囲メッセージ (デフォルトまたはユーザー指定) を定義すると、そのメッセージが表示されます。

日付　すべてのロケールで、日付は MM/DD/YYYY 形式で入力します。

デフォルト値

オプション。このフィールドの値は、ホスト・アプリケーションで関連の入力フィールドに、有効な日付 (指定された日付制限の間の日付) が事前に設定されなかった場合、ポップアップ・カレンダーで選択される初期日付です。このフィールドと関連するホスト入力フィールドとの間の相互作用は、次のように処理されます。

- ホスト・アプリケーションにより入力フィールドにゼロまたは不適切な日付形式が事前に設定されると、このホスト・フィールドのデータはデフォルト値によって上書きされません。関連の入力フィールドを更新するには、ユーザーが手動で日付を入力するか、またはポップアップ・カレンダーから日付を選択する必要があります。
- デフォルト値が指定されていないか、またはデフォルト値のフォーマットが誤っている場合、本日の日付が最初にポップアップ・カレンダーで選択されます。
- 関連の入力フィールドにデータを事前に設定するには、変換を適用する前に、画面カスタマイズ・イベントに「**データを挿入**」アクションを追加します。

注: すべてのロケールで、日付パターンは MM/DD/YYYY です。

デフォルト無効メッセージの使用

無効な日付をユーザーが入力したときにデフォルトの Dojo 無効メッセージを表示するには、このボックスを選択します。表示する独自の無効メッセージを指定するには、このボックスをクリアします。

無効メッセージ

独自の無効メッセージを指定するには、このフィールドを使用します。このフィールドを空のままにすると、メッセージは表示されず、単に入力フィールドが強調表示されます。

注: 独自の無効メッセージを指定すると、ユーザーにはその指定した言語で表示されます。翻訳は行われません。

デフォルト範囲メッセージの使用

ユーザーが範囲外の日付を入力したときにデフォルトの Dojo 範囲メッセージを表示するには、このボックスを選択します。表示する独自の範囲メッセージを指定するには、このボックスをクリアします。

範囲メッセージ

独自の範囲メッセージを指定するには、このフィールドを使用します。このフィールドを空のままにすると、メッセージは表示されず、単に入力フィールドが強調表示されます。

注: 独自の範囲メッセージを指定すると、ユーザーにはその指定した言語で表示されます。翻訳は行われません。

プロンプト・メッセージ

ユーザーがテキスト・ボックスを選択した際に表示するプロンプト・メッセージを指定するには、このフィールドを使用します。例えば、使用する正しい日付形式をユーザーに知らせたり、ポップアップ・カレンダーから日付を選択するように注意したりできます。

注: 独自のプロンプト・メッセージを指定すると、ユーザーにはその指定した言語で表示されます。翻訳は行われません。

サブミット時に再検証

無効な日付をユーザーが入力したときにページ・サブミットが行われないようにするには、このボックスを選択します。このボックスが選択されていない場合、無効な日付にはフラグが立ちますが、その日付は依然としてホストにサブミットできます。

拡張グリッド (Dojo) Web のみ

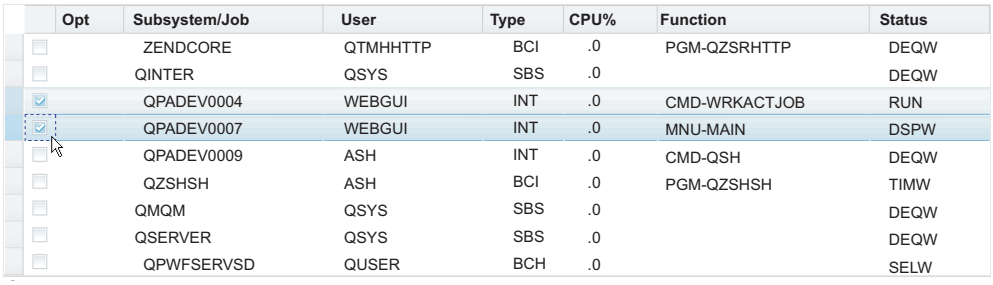
拡張グリッド・ウィジェットは、Dojo テーブルのレンダリングを担当します。レンダリングされるテーブルは読み取り専用です。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- テーブル
- テーブル (フィールド)
- テーブル (可視)

次の図は、変換時に拡張グリッド・ウィジェットがどのように表示されるのかを示したものです。 テーブル・コンポーネントの例に示されたデータと同様のデータが入力として使用されています。

注: この図は、Dojo Claro テーマを使用した場合のウィジェットの外観を示しています。異なる Dojo テーマを使用した場合のウィジェットの外観は、これとは異なることがあります。



Opt	Subsystem/Job	User	Type	CPU%	Function	Status
	ZENDCORE	QTMHHTTP	BCI	.0	PGM-QZSRHTTP	DEQW
	QINTER	QSYS	SBS	.0		DEQW
☑	QPADEV0004	WEBGUI	INT	.0	CMD-WRKACTJOB	RUN
☑	QPADEV0007	WEBGUI	INT	.0	MNU-MAIN	DSPW
	QPADEV0009	ASH	INT	.0	CMD-QSH	DEQW
	QZSHSH	ASH	BCI	.0	PGM-QZSHSH	TIMW
	QMQM	QSYS	SBS	.0		DEQW
	QSERVER	QSYS	SBS	.0		DEQW
	QPWFSERVSD	QUSER	BCH	.0		SELW

図 63. Dojo 拡張グリッド・ウィジェットの例

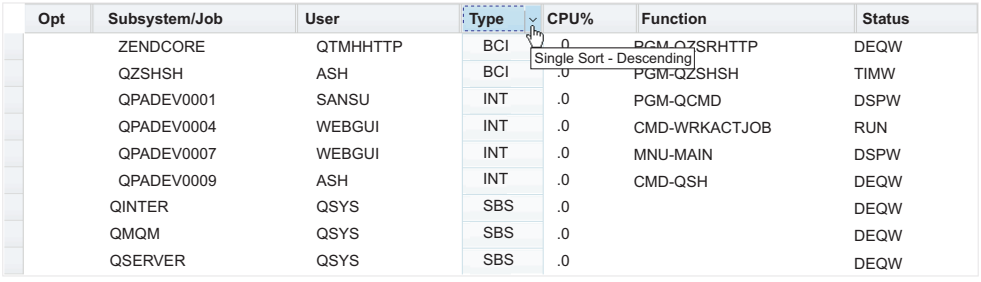
Dojo Claro テーマを使用しているとき、拡張グリッド・ウィジェットのデフォルト・レンダリングには、テーブルの最初の列として行セレクター列が含まれます。ユーザーは、この列のセルをクリックして、テーブル内の別の場所にドラッグする行を選択できます。複数の連続する行を選択するには、最初の行をクリックして、Shift を押して最後の行をクリックするか、最初の行をクリックしてマウスを押しながら、最後の行まで進みます。連続しない複数の行を選択するには、Ctrl を押しながら各行をクリックします。ドラッグするには、選択した行をクリックしてドラッグします。

注: テーブルの 1 つのセルまたは行にフォーカスが置かれている場合は、Enter キーを押すと、ウィジェットによって処理され、ホスト・アプリケーションには送信されません。

拡張グリッド・ウィジェットのデフォルト・レンダリングには、別の機能としてソート機能が含まれています。行を列の内容で昇順または降順にソートすること、およびソートを解除することができます。ネスト・ソート、つまりソートされた最初の列内でネストされた 2 番目の列のソートを実行することもできます。

注: 列のソートは、単純なストリングの比較に基づいています。そのため、一部のストリング (例えば、`mm/dd/yy` 形式の日付ストリング) は予期したとおりにソートされません。

ソートを実行するには、以下の例に示すように列見出しをクリックします。



Opt	Subsystem/Job	User	Type	CPU%	Function	Status
	ZENDCORE	QTMHHTTP	BCI	.0	PGM-QZSRHTTP	DEQW
	QZSHSH	ASH	BCI	.0	PGM-QZSHSH	TIMW
	QPADEV0001	SANSU	INT	.0	PGM-QCMD	DSPW
	QPADEV0004	WEBGUI	INT	.0	CMD-WRKACTJOB	RUN
	QPADEV0007	WEBGUI	INT	.0	MNU-MAIN	DSPW
	QPADEV0009	ASH	INT	.0	CMD-QSH	DEQW
	QINTER	QSYS	SBS	.0		DEQW
	QMQM	QSYS	SBS	.0		DEQW
	QSERVER	QSYS	SBS	.0		DEQW

Single Sort - Descending

図 64. Dojo 拡張グリッド・ウィジェットのソートの例

フィルター操作選択 (Dojo) Web のみ

フィルター操作選択ウィジェットは、Dojo ドロップダウン入力テキスト・ボックスのレンダリングを担当します。

フィルター操作選択ウィジェットの特性は次のとおりです。

- ホストに実行依頼するリスト項目の表題と対応するリスト項目値で構成される値の組のリストが、ドロップダウン入力テキスト・ボックスに表示されます。レンダリングされるコンポーネントによっては、値の組をコンポーネントによって指定するか、ウィジェット設定でユーザーが手動で指定することができます。

  注: 表題と値の両方がドロップダウン・リストに表示されます。
- フィルター操作選択ウィジェットは編集可能であるため、ユーザーは、実行時に値を入力できます。
- ユーザーが入力テキスト・ボックスに入力するときに、リストで部分的に一致する値がドロップダウン・リストに表示されます。
- リストの値と一致しないユーザー指定のテキストには、エラーのフラグが立てられます。無効な入力が行われた結果、ブランクがホストに実行依頼されます。

  注: フィルター操作選択ウィジェットとコンボ・ボックス・ウィジェットの主な違いは、フィルター操作選択ウィジェットではユーザーは、既にリストにない値をサブミットできないことです。
- ウィジェットは、多くの API オプションを使用してカスタマイズ可能です。そのようなオプションには、検証および制約オプション、エラー・メッセージ、アイコン、オートコンプリートなどが含まれます。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- コマンド行
- 入力フィールド
- ヒント付き入力フィールド
- 項目の選択
- 選択リスト

次の図に、変換において選択リスト・コンポーネントの例のデータを入力として使用するフィルター操作選択ウィジェットの例を示します。

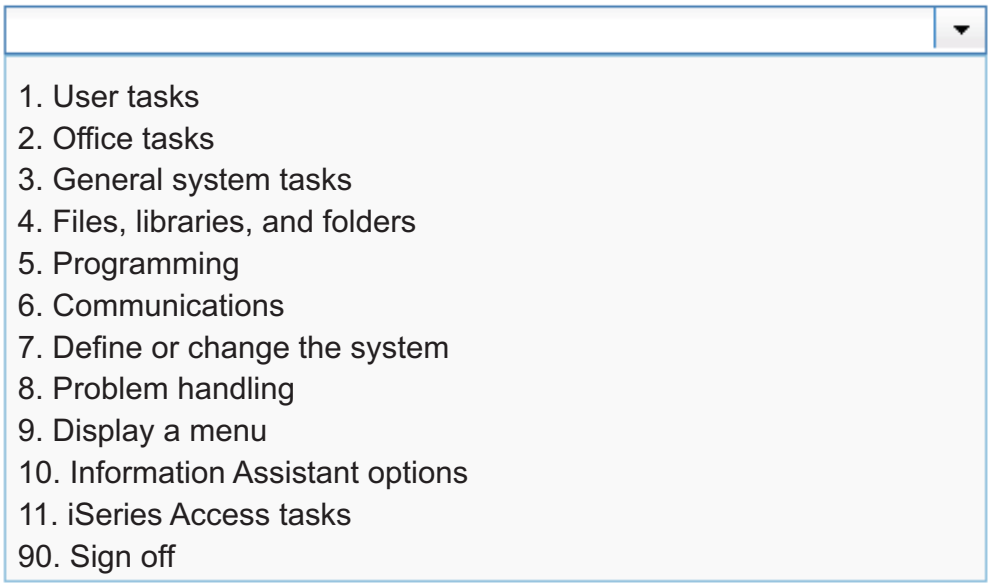


図 65. Dojo フィルター操作選択ウィジェットの例

以下の図は、ポップアップ・リストに表示される、部分的に一致した値の例を示しています。この例では、ユーザーは入力テキスト・ボックスに 1 と入力しました。

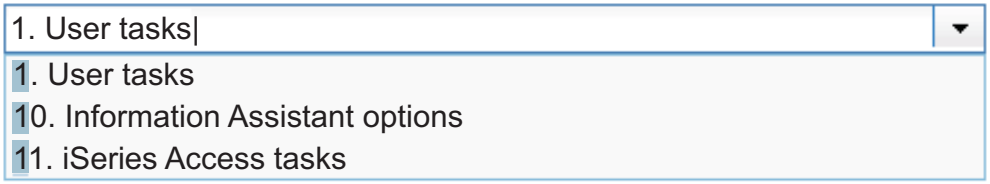


図 66. Dojo フィルター操作選択ウィジェットのフィルター操作の例

以下の図に示すように、フィルター操作選択ウィジェットでは、コンボ・ボックス・ウィジェットと異なり、ユーザーは表示されるリストにない値を入力して実行依頼することはできません。

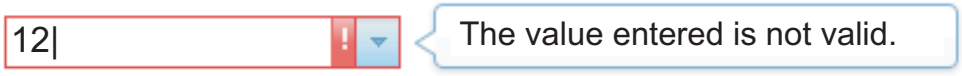


図 67. Dojo フィルター操作選択ウィジェットで無効なユーザー指定入力が行われた例

以下の設定が、このウィジェットに対して構成できます。

グローバル変数から設定
: これを選択した場合、指定したグローバル変数がドロップダウン・リストに設定されます。

リストの値を含むグローバル変数

値の集合を含む索引付きグローバル変数の名前を指定します。グローバル変数内の各索引の項目がドロップダウン・リストに作成されます。

共用 HATS のローカル・グローバル変数と共用グローバル変数は同じ名前を持つことができます。共用グローバル変数を使用してリスト項目にデータを取り込む場合は、このボックスを選択します。このボックスをクリアした場合は、ローカル・グローバル変数が使用されます。

リストの表題を含むグローバル変数

オプション。表題の集合を含む索引付きグローバル変数の名前を指定します。この値によって指定されるグローバル変数のサイズは、前の設定で指定されたサイズ以上である必要があります。この索引付きグローバル変数の索引は、グローバル変数の値の索引と一致する必要もあります (その結果、ユーザーに表示される実際の値と表題が同期します)。この値が指定されない場合は、ドロップダウン・リスト内の各項目の表題がその値になります。

ストリングから設定

これを選択した場合、指定したストリングがリストに設定されます。

リスト項目

オプション。ドロップダウン・リストに含まれる項目のストリングを指定します。項目はセミコロン (;) で分離する必要があります。リスト項目の表題とリスト項目の値が異なるようにするには、表題と値を等号 (=) で区切って両方を入力します。例えば、**Apple=A;Grape=G** の値は、2 つの項目「**A=Apple**」と「**G=Grape**」を持つドロップダウン・リストをレンダリングします。最初の項目を選択すると、関連するホスト画面の入力フィールドに「**A**」が挿入されます。

ドロップダウン・リストの項目とホスト画面に挿入される値を同一にするには、項目のみを入力します。例えば、**Apple=A;G** です。この例では、ドロップダウン・リストとホスト画面の入力フィールドに **G** が表示されます。

選択時に自動的に実行依頼

これを選択した場合、ドロップダウン・リストで選択操作が行われると、その選択内容が実行依頼されます。

テキスト・ボックス (Dojo) Web のみ

テキスト・ボックス・ウィジェットは、Dojo テキスト・ボックスのレンダリングを担当します。

このウィジェットは、以下のコンポーネントが提供するデータをレンダリングします。

- コマンド行
- 入力フィールド

次の図に、変換においてコマンド行コンポーネントの例のデータを入力として使用するテキスト・ボックス・ウィジェットの表示例を示します。

図 68. Dojo テキスト・ボックス・ウィジェットの例

注: 入力フィールドの上にあるラベルを表示する場合は、ソースにおいてそのフィールドが含まれている <td> タグで幅を調整すれば、そのフィールドを希望どおりにレンダリングできます。

検証テキスト・ボックス (Dojo) Web のみ

検証テキスト・ボックス・ウィジェットは、指定された正規表現を使用してユーザー入力を検証する Dojo テキスト・ボックスのレンダリングを担当します。

このウィジェットは、以下のコンポーネントの提供するデータをレンダリングします。

- 入力フィールド

次の図に、変換において指定されたオプションのプロンプト・メッセージを使用する検証テキスト・ボックス・ウィジェットの表示例を示します。

図 69. プロンプト・メッセージが表示された Dojo 検証テキスト・ボックス・ウィジェットの例

次の図は、有効な入力が行われた検証テキスト・ボックス・ウィジェットを示しています。

図 70. 有効な入力が行われた Dojo 検証テキスト・ボックス・ウィジェットの例

次の図は、デフォルトの無効メッセージが表示された検証テキスト・ボックス・ウィジェットを示しています。

図 71. 無効メッセージが表示された Dojo 検証テキスト・ボックス・ウィジェットの例

以下の設定が、このウィジェットに対して構成できます。

正規表現

ユーザー指定のデータの形式を検証する正規表現を指定するには、このフィールドを使用します。このフィールドが空の場合、検証は実行されず、ユーザーは自由に入力できます。

検証 「**正規表現**」フィールドの正規表現の構文を検証するには、このボタンをクリックします。

プロンプト・メッセージ

ユーザーがテキスト・ボックスを選択した際に表示するプロンプト・メッセージを指定するには、このフィールドを使用します。例えば、「正規表現」フィールドで定義された正しいデータ形式をユーザーに知らせることができます。

注: 独自のプロンプト・メッセージを指定すると、ユーザーにはその指定した言語で表示されます。翻訳は行われません。

デフォルト無効メッセージの使用

無効なデータをユーザーが入力したときにデフォルトの Dojo 無効メッセージを表示するには、このボックスを選択します。表示する独自の無効メッセージを指定するには、このボックスをクリアします。

無効メッセージ

独自の無効メッセージを指定するには、このフィールドを使用します。このフィールドを空のままにすると、メッセージは表示されず、単に入力フィールドが強調表示されます。

注: 独自の無効メッセージを指定すると、ユーザーにはその指定した言語で表示されます。翻訳は行われません。

サブミット時に再検証

無効なデータをユーザーが入力したときにページ・サブミットが行われないようにするには、このボックスを選択します。このボックスが選択されていない場合、無効なデータにはフラグが立ちますが、そのデータは依然としてホストにサブミットできます。

コンポーネントおよびウィジェットのマッピング

GUI でホスト・コンポーネントを表示するときに使用される HATS Toolkit のウィジェットは、それぞれのコンポーネントにマップされています。次の表は、既存の HATS ホスト・コンポーネントおよび対応ウィジェットのリストです。

表 2. HATS ホスト・コンポーネントおよび対応するウィジェット

ホスト・コンポーネント	ウィジェット
コマンド行	コンボ RCP のみ ドロップダウン (データ入力) リスト ポップアップ ラジオ・ボタン (データ入力) テキスト入力 コンボ・ボックス (Dojo) Web のみ フィルター操作選択 (Dojo) Web のみ テキスト・ボックス (Dojo) Web のみ
フィールド	フィールド
ファンクション・キー	ボタン ボタン・テーブル ドロップダウン (選択) リンク ラジオ・ボタン (選択) ツールバー RCP のみ
HTML DDS キーワード	ラベル
入力フィールド	カレンダー Web のみ チェック・ボックス コンボ RCP のみ ドロップダウン (データ入力) リスト ポップアップ ラジオ・ボタン (データ入力) テキスト入力 コンボ・ボックス (Dojo) Web のみ 日付テキスト・ボックス (Dojo) Web のみ フィルター操作選択 (Dojo) Web のみ テキスト・ボックス (Dojo) Web のみ 検証テキスト・ボックス (Dojo) Web のみ
ヒント付き入力フィールド	コンボ RCP のみ ドロップダウン (データ入力) リスト ポップアップ ラジオ・ボタン (データ入力) コンボ・ボックス (Dojo) Web のみ フィルター操作選択 (Dojo) Web のみ

表 2. *HATS* ホスト・コンポーネントおよび対応するウィジェット *(続き)*

ホスト・コンポーネント	ウィジェット
項目の選択	チェック・ボックス コンボ RCP のみ ドロップダウン (データ入力) リンク (項目選択) リスト ポップアップ ラジオ・ボタン (項目選択) テキスト入力 コンボ・ボックス (Dojo) Web のみ フィルター操作選択 (Dojo) Web のみ
ライト・ペン (アテンション)	ボタン リンク ラジオ・ボタン (選択)
ライト・ペン (選択)	チェック・ボックス
スクロール・バー・フィールド (ENPTUI)	スクロール・バー (ENPTUI)
選択フィールド (ENPTUI)	ボタン チェック・ボックス リンク ラジオ・ボタン (選択)
選択リスト	ボタン ボタン・テーブル ドロップダウン (選択) リンク ラジオ・ボタン (選択) ツールバー RCP のみ コンボ・ボックス (Dojo) Web のみ フィルター操作選択 (Dojo) Web のみ
サブファイル	サブファイル (チェック・ボックス) サブファイル (ドロップダウン) サブファイル (ポップアップ)
テーブル	グラフ (水平棒) グラフ (線) グラフ (垂直棒) テーブル 拡張グリッド (Dojo) Web のみ
テーブル (フィールド)	グラフ (水平棒) グラフ (線) グラフ (垂直棒) テーブル 拡張グリッド (Dojo) Web のみ
テーブル (可視)	グラフ (水平棒) グラフ (線) グラフ (垂直棒) テーブル 拡張グリッド (Dojo) Web のみ
テキスト	ラベル

表 2. *HATS* ホスト・コンポーネントおよび対応するウィジェット *(続き)*

ホスト・コンポーネント	ウィジェット
URL	リンク

第 10 章 テンプレートの使用

テンプレートを使用すると、会社の Web サイトやアプリケーションのスタイル、またはその他のガイドラインに一致するようにテンプレートをカスタマイズすることにより、HATS アプリケーションの外観を細かく制御できます。テンプレートとは、HATS Web プロジェクトでの JSP ファイル、または HATS リッチ・クライアント・プロジェクトでの Java SWT コンポジット・クラスのことです。テンプレートは、HATS 変換によってレンダリングされるホスト画面用に予約されている領域です。テンプレートには、会社のロゴや情報を収容できます。HATS Web テンプレートの場合には、他の Web ページへのリンクも組み込むことができます。また、テンプレートは、変換されたホスト画面が表示されるエリアの背景色、イメージ、その両方を定義します。

「HATS テンプレート」ウィザードを起動すると、ブランク・テンプレート、または HATS に付属しているいずれかのテンプレートに基づくテンプレートを作成できます。HATS Web プロジェクトの場合、テンプレートを Web ページ (URL またはファイル) に基づかせることもできます。既存の Web ページを会社の Web サイトに基づくテンプレートとしてインポートすることもできます。HATS リッチ・クライアント・プロジェクトの場合は、Java ソース・ファイルをプロジェクトにコピーするか、Eclipse インポート・ウィザードを使用すれば、テンプレートを Java ソース・ファイルとして他のプロジェクトからインポートできます。

HATS は、ユーザーのプロジェクトに使用できるテンプレートを備えています。これらのテンプレートの名前を表示するには、「**Web コンテンツ**」 Web のみ フォルダーまたは「**リッチ・クライアント・コンテンツ**」 RCP のみ フォルダーを展開し、「**HATS プロジェクト**」ビューの「**テンプレート**」フォルダーを選択します。付属のテンプレートには、以下の定義の一部またはすべてが組み込まれています。

- デフォルトの背景色
- デフォルトの前景色
- ホストの色を GUI の色に対応付けるためのデフォルトの色マッピング
- デフォルトのフォント
- 変換によりレンダリングされたホスト画面の領域
- Web テンプレートの場合
  - ウィジェットのスタイル属性
  - **Web Content/Common/Images** フォルダーに置かれている .gif ファイル、.jpg ファイル、および .png ファイルを使用して作成されるイメージ定義
  - HATS のデフォルト・アプリケーション・キーパッド

注:

1. プロジェクト設定エディターの「**テンプレート**」タブでは、新規に提供されたテンプレートには、アスタリスクでマークが付けられています。
2. HATS プロジェクトを作成し、「**モバイル装置用にオプションを最適化**」を選択したとき、モバイル装置用に最適化されたテンプレートのみが、プロジェクトでの使用のために提供されます。

© Copyright IBM Corp. 2003, 2015

テンプレートの例

このセクションでは、HATS に付属の一部のテンプレートについて説明します。

図 72 は、HATS Web アプリケーションで使用するために付属している finance.jsp テンプレートを使用したホスト画面を示しています。このテンプレートは、他の Web サイトへのリンクを持つドロップダウン・メニュー、および下部のアプリケーション・キーパッドを使用する例を提供します。

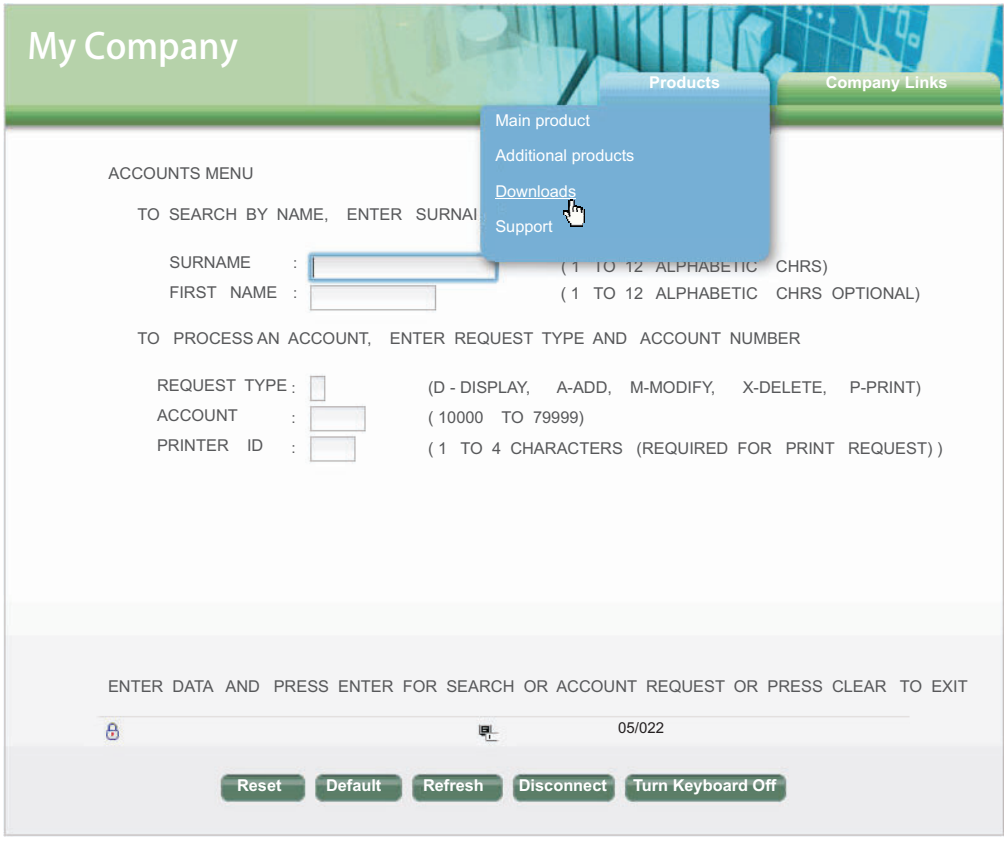


図 72. HATS 提供の Web アプリケーション finance テンプレート

369 ページの図 73 は、HATS Web アプリケーションで使用するために付属している industry.jsp テンプレートを使用したホスト画面を示しています。このテンプレートは、他の Web サイトへのメニュー・バー・リンク、および下部のアプリケーション・キーパッドを使用する例を提供します。

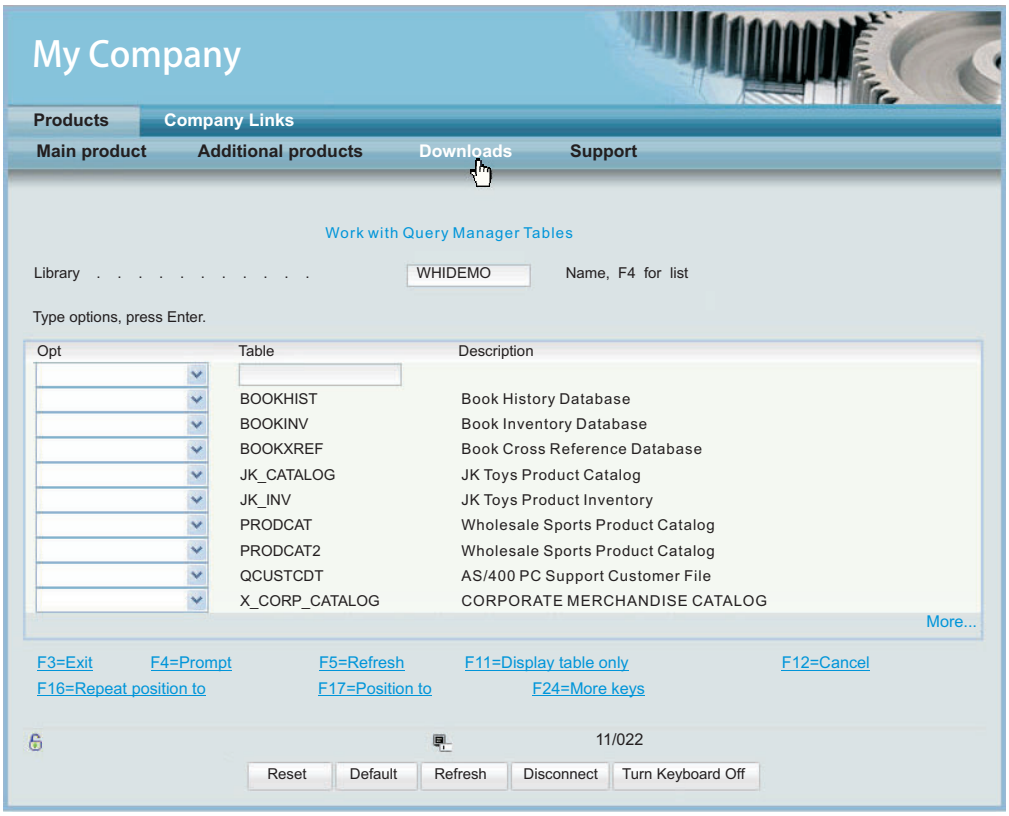


図 73. HATS 提供の Web アプリケーション industry テンプレート

370 ページの図 74 は、HATS リッチ・クライアント・アプリケーションで使用するために付属している modern.java テンプレートを使用したホスト画面を示しています。このテンプレートは、ボタン、リンクなどの全体的に共通の外観をランタイム環境 (例えば、Eclipse RCP、Lotus Notes、Lotus Expeditor など) が提供するという前提に基づいて、比較的簡単な、サイドバーのみの構成になっています。

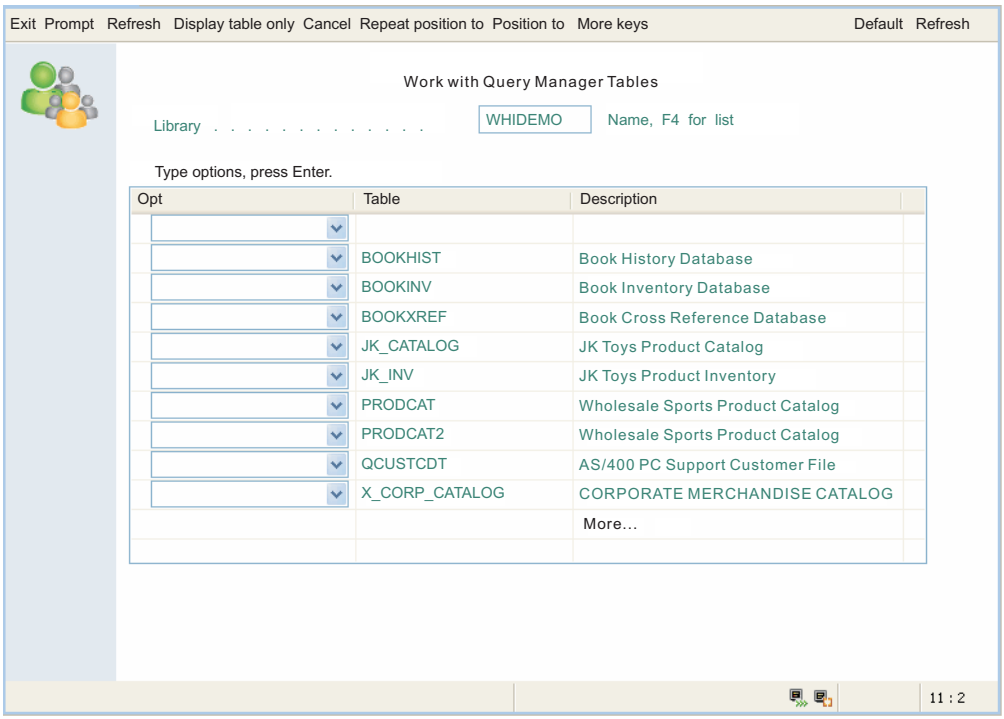


図 74. HATS 提供の リッチ・クライアント・アプリケーション modern テンプレート

「**HATS Projects**」ビューで HATS リソースのコピーを作成するには、任意のフォルダーを開いてリソースを右クリックし、「**コピー**」を選択します。 リソースの「**貼り付け**」先フォルダーを選択すると、HATS から名前の入力を求められます。これは個人用テンプレートの作成を開始するときのバックアップとして役立ちます。

注: 「**HATS Projects**」ビューから「**ナビゲーター**」ビューにコピーすることはできません。

HATS Toolkit でプロジェクトを作成する場合、プロジェクトのデフォルト・テンプレートとして使用するテンプレートを選択します。このテンプレートは「**HATS Projects**」ビューに異なるアイコンで表示され、このアイコンの横には「**デフォルト**」という文字が表示されます。デフォルト・テンプレートを変更するには、「**HATS Projects**」ビューに移動し、**Web Content/Templates** Web のみ、または **Rich Client Content/Template** RCP のみ に置かれているテンプレートを右クリックして、「**デフォルト・テンプレートとして設定**」を選択します。 もう 1 つの方法では、「**HATS Projects**」ビューで、HATS プロジェクトの「**プロジェクト設定**」をダブルクリックします。「**テンプレート**」タブに移動し、リストで選択します。

画面カスタマイズを作成するときに、アクションが変換の適用または URL の表示である場合、変換の適用時に使用するテンプレートを選択できます。ホスト・アプリケーションがすべての画面で一貫性を保つようにするには、プロジェクトのデフォルト・テンプレートとして選択したテンプレートを変換にも使用します。これを行うには、このプロジェクトのデフォルト・テンプレートとして使用するテンプレートを選択します。

「テンプレートの作成」ウィザード

個人が使用する機能や会社のスタイル・ガイドラインに合った個人用のカスタム・テンプレートを作成できます。プロジェクト用のカスタム・テンプレートは、HATS Toolkit 内でウィザードとエディターを使用してデザインできます。HATS は、作成するすべてのテンプレートに、色およびフォント、HATS のデフォルト・アプリケーション・キーパッド、および変換でレンダリングされるホスト画面用の領域を含めるのに必要なコードを自動的に追加します。

「テンプレートの作成」ウィザードを使用して、新規テンプレートを作成します。ウィザードにアクセスするには、以下のようにさまざまな方法があります。

- 「**HATS Projects**」ビューで、プロジェクトを右クリックして、「**新規 HATS**」>「**テンプレート**」を選択します。
- HATS ツールバーの「**HATS テンプレートを作成**」アイコンをクリックします。
- HATS メニュー・バーで、「**HATS**」>「**新規**」>「**テンプレート**」 (または「**ファイル**」>「**新規**」>「**HATS テンプレート**」) を選択します。

ウィザードで、プロジェクトを指定して、テンプレートに名前を付けます。テンプレートがリッチ・クライアント・プロジェクト用である場合、「**パッケージ**」フィールドには、デフォルトの Java パッケージの HATS プリファレンスで規定されているデフォルトのパッケージ名が事前に入力されています。詳細については、 149 ページの『HATS プリファレンスの使用』を参照してください。「**参照**」ボタンを使用すると、別のパッケージを探して選択できます。必要に応じて、テンプレートに説明を設定し、「**次へ**」をクリックして、以下のいずれかのオプションを選択します。

ブランク・テンプレートの作成

このオプションを選択して、ブランク・テンプレートを作成します。

既存のプロジェクト・テンプレートからテンプレートを事前設定

既存のテンプレートを基にして新規のテンプレートを事前に設定するには、このオプションを選択します。既存のテンプレートの名前を選択します。実際には、この操作によって「テンプレートの作成」ウィザードの始動時に指定した名前を使用して、既存のテンプレートのコピーが作成されます。下のウィンドウに選択したテンプレートのプレビューが表示されます。

既存の Web ページ (URL またはファイル) からテンプレートを事前設定 Web のみ

既存の Web ページ (会社の既存の Web サイトなど) または特定のファイルを基にして新規のテンプレートを事前に設定するには、このオプションを選択します。テキスト・ボックスに URL またはファイル名を入力するか、「**参照**」ボタンをクリックして検索します。「**プレビュー**」をクリックすると、選択したテンプレートが表示されます。

Web ページのほか、リンクされたファイル (例: イメージ、スタイル・シート) が Web からダウンロードされます。ファイルへのリンクは、ユーザーの HATS プロジェクト内の対応するパスに合わせて変更されます。

注: インポートした Web ページにカスタム JavaScript、アプレットなどが埋め込まれている場合は、問題が生じる可能性があります。ご使用の Web サイトに複雑なタグが含まれている場合は、テンプレートが HATS と連携するようにそれらのタグを変更する必要が生じる場合があります。「テンプレートの作成」ウィザードでは、該当のページ、すべてのイメージ、およびスタイル・シートがダウンロードされますが、サポートされていない可能性がある JavaScript 呼び出しのフィルターによる除去は試行されません。フレームを使用する Web ページはサポートされません。

新規テンプレートをプロジェクトのデフォルト・テンプレートにするには、「**このプロジェクトのデフォルト・テンプレートとして設定**」チェック・ボックスを選択します。

「**完了**」をクリックします。これにより、新規のテンプレートに変更や修正を加えることができる適切なエディターが起動します。

プロジェクト内のオブジェクトに加える変更は、そのプロジェクトにのみ影響するので注意してください。別のプロジェクト用に作成したテンプレートを使用する場合は、そのテンプレートを、変更したスタイル・シートまたはイメージとともに、作成する新規プロジェクトにコピーする必要があります。

HATS Web プロジェクトで、独自のテンプレートを HATS Toolkit 環境の外部に作成する場合は、以下のステートメントを組み込んであることを確認してください。

<LINK rel="stylesheet" href="xxx">

次の各設定を定義するスタイル・シートを参照する 1 つ以上のタグ。

- ボタン、リンク、およびキーパッドの外観
- テンプレートの背景色およびウィジェット
- テキストのフォントとサイズ

<HATS:Transform>

このタグは、テンプレートで囲む変換対象位置を定義します。

テンプレートの編集

HATS Web プロジェクト内のテンプレートは JSP ファイルで、デフォルトでは Page Designerを使用して編集します。 HATS リッチ・クライアント・プロジェクト内のテンプレートは Java SWT コンポジットで、Java ビジュアル・エディター・フィーチャーが Rational SDP とともにインストールされていると想定すると、デフォルトでは Visual Editor を使用して編集されます。

Web プロジェクトのテンプレートの編集

すべてのテンプレート (HATS に付属のテンプレートと作成したテンプレート) を参照するには、「**HATS Projects**」ビューの「**Web コンテンツ**」の下にある「**テンプレート**」フォルダーを展開します。 Rational SDP に組み込まれている Page Designer を使用してテンプレートを編集するには、テンプレートの名前をダブルクリックします。使用可能なその他のエディターを参照するには、テンプレートの名前を右クリックして、「**オープン先を指定**」を選択します。 メニュー・バーの「**へ**

ルプ」>「**ヘルプ目次**」を選択して「Page Designer」を検索することにより、Page Designerに関する Rational SDP 資料を参照してください。

テンプレートの編集には、次の制限が適用されます。

- カスタム・テンプレートは UTF-8 でエンコードされている必要があります。
- 名前が HATS、hats、または Hats で始まる JSP 変数、CSS クラス、HATSForm、またはその他のオブジェクトを使用しないでください。これらの名前は HATS 用に予約されています。
- HATS_form というフォーム名を作成しないでください。HATS が、HTML 出力をアセンブルするときにこのフォームを作成します。
- HATS:Transform タグを囲むフォームを使用しないでください。HATS が、HATS:Transform タグが存在する位置にフォームを作成します。HTML では、ネストされたフォームを使用できません。

次のセクションでは、Page Designer の各タブを説明します。

デザイン

テンプレートを変更すると、Page Designerの「**デザイン**」タブには、そのテンプレートの現在のビューが表示されます。このタブでは、Rational SDP ツールバーから追加の編集オプションが使用できます。例えば、ツールバーから「**挿入**」ドロップダウン・メニューを使用して、イメージ、写真、テキスト、テキスト・フォーマット制御などを挿入したり、「**パレット**」から HATS コンポーネントをドラッグすることができます。テンプレートが表示されたときに演奏される音楽の追加、レイアウト・フレームの作成、および HTML タグの追加を行うことができます。

グローバル変数、マクロ・キー、ホストおよびアプリケーション・キーパッド、または個別のキーを挿入することもできます。HATS Toolkit メニュー・バーの「**HATS ツール**」メニューを使用すると、これらの項目を挿入できます。

イメージをプロジェクトに追加する場合は、イメージをプロジェクトの Web Content/Common/Images ディレクトリーにインポートしてください。イメージをインポートするには、「**ファイル**」>「**インポート**」>「**一般**」>「**ファイル・システム**」をクリックして「インポート」ウィザードを開きます。「**ソース・ディレクトリー (From directory)**」フィールドで、インポートするイメージ・ソース・ファイルの場所を選択します。*project_name*/Web Content/common/images ディレクトリーを、宛先となる「**ターゲット・フォルダー (Into Folder)**」として選択します。イメージ・ソース・ファイルがインポートされたら、**Images** フォルダーを右マウス・ボタンでクリックし、右下ウィンドウの「**サムネール (Thumbnail)**」タブで「**サムネールを表示**」を選択して、フォルダー内のイメージを表示します。イメージは、テンプレートの「**デザイン**」タブ表示にドラッグ・アンド・ドロップしてコピーできます。

注: GB18030 各国語文字を使用する場合の考慮事項については、510 ページの『コード・ページ 1388 (GB18030) の使用』を参照してください。

「**HATS ツール**」メニューの「**ホスト・コンポーネントを挿入**」、「**マクロの挿入**」、または「**グローバル変数を挿入**」項目をクリックすると、変換時にそれらの項目を定義する場所にウィザードが現れます。これらのウィザードの詳細について

は、 213 ページの『ホスト・コンポーネントを挿入』、 217 ページの『マクロ・キーを挿入』、および 218 ページの『グローバル変数を挿入』を参照してください。

スタイル・シートの使用: Cascading Style Sheet とは、HTML のレンダリングを制御するスタイル規則の階層セットを定義するファイルです。このファイルを使用すると、HATS によりレンダリングされた画面領域とテンプレートのスタイルとの整合性を維持するために、フォント色、サイズ、背景色などの出力要素を制御できます。

スタイル・シートの変更: HATS は、カラー・スキームとフォント・サイズを変更するスタイル・シートを備えています。これらのスタイル・シートの少なくとも 1 つがテンプレートに適用されます。「**デザイン**」タブでテンプレートを表示して、これらのスタイル・シートをテンプレートに適用できます。

「**HATS Projects**」ビューの **Web Content/Common/Stylesheets** フォルダーに移動すると、編集するスタイル・シートをダブルクリックできます。デフォルトでは、これで CSS Designer エディター・ウィンドウおよびプレビュー・ウィンドウにスタイル・シートが開かれ、スタイル・シート内のスタイルのサンプルが表示されます。

CSS Designer エディター・ウィンドウ内の特定のスタイルをクリックすると、スタイルのサンプルがプレビュー・ウィンドウに表示されます。

編集するスタイルの判別: ホスト画面のフィールドの色を変更したいことはよくあります。問題は、その特定のホスト・フィールドに適用されているスタイルを確認することです。これを判別するには、ホスト・フィールドが表示されているときに Web ページのソースを表示し、そのソースを調べることにより、適用されているスタイルを判別できます。

HATS プロジェクトのサインオン画面の「ユーザー」および「パスワード」テキストに使用されているテキストを変更するとします。Web ページのソースを表示するには、実際の Web ブラウザーで該当のフィールドが表示されるまで HATS プロジェクトを実行します。Rational SDP に組み込まれている Web ブラウザーは使用しないでください。Rational SDP のテスト・サーバー上で HATS プロジェクトを実行することはできますが、プロジェクトにアクセスするには実際の Web ブラウザー (例えば Internet Explorer) を使用する必要があります。

該当のフィールドが表示されたら、例えば Internet Explorer を使用している場合は、ブラウザーのメニュー・バーにある「**表示**」をクリックし、ドロップダウン・メニューから「**ソース**」をクリックします。

テキスト・ファイルで、変更するフィールドまたはそれに近いフィールド内のテキスト (この例では、「**ユーザー**」) を検索します。 ソース内のテキストの直前に、`CLASS=` ステートメントがあります。このステートメントは、テキストに適用されている CLASS またはスタイルを規定します。そこで、テンプレートまたはスタイル・シートに移動してスタイルを編集すると、希望する効果を得られます。

注: ソースを表示する代わりに、ご使用のブラウザーで使用可能であれば、新規の開発者ツール (例えば Internet Explorer 開発者ツールなど) を使用して、特定のページ要素を検査できます。

スタイル変更は、このスタイルが使用される HATS アプリケーション全体に適用されます。デフォルト・アプリケーションを作成する場合、そのアプリケーションではすべてのホスト画面にデフォルト・テンプレートが適用され、すべての画面でこのスタイルが使用されます。スタイルを 1 つのホスト画面のみに適用したい場合、その特定のホスト画面に適用するテンプレートを作成する必要があります。

Cascading Style Sheet の詳細については、http://www.w3.org/Style/CSS/ を参照してください。

HATS スタイル・シート: HATS には以下のスタイル・シートが付属し、作成する各新規プロジェクトに組み込まれます。

表 3. HATS スタイル・シート

名前	説明
blacktheme	黒の背景。前景色はホスト画面の色と一致します。
calendar	カレンダー・ポップアップのスタイル
commontheme	すべてのスタイル・シートに共通の要素を定義します。メイン・スタイル・シート (blacktheme、 graytheme、 monochrometheme、 tantheme、 whitetheme) によりインポートされます。
finance	Finance.jsp テンプレート用のスタイル・シート
graytheme	グレーの背景。前景色はさまざまな色調のグレーです。
industry	Industry.jsp テンプレート用のスタイル・シート
inlineCalendar	カレンダー・ウィジェットの inlineCalendar オプション用のスタイル・シート
largeFont	ブラウザー設定およびクラス関数に関連して生成されたすべての出力のフォント・サイズ
medical	Medical.jsp テンプレート用のスタイル・シート
monochrometheme	白の背景。前景色はさまざまな色調のグレー、リンクは青です。
nonFixedFont	白の背景。フォント・サイズはさまざまです。
normalFont	ブラウザー設定およびクラス関数に関連して生成されたすべての出力のフォント・サイズ
PrintJobWindow	「印刷ジョブ」ウィンドウのスタイル
research	Research.jsp テンプレート用のスタイル・シート
reverseVideoBlack	blacktheme スタイル・シートの前景色と背景色を反転します。
reverseVideoGray	graytheme スタイル・シートの前景色と背景色を反転します。
reverseVideoMono	monochrometheme スタイル・シートの前景色と背景色を反転します。
reverseVideoTan	tantheme スタイル・シートの前景色と背景色を反転します。
reverseVideoWhite	whitetheme スタイル・シートの前景色と背景色を反転します。
scaleableFont	生成されたすべての出力のフォント・サイズはブラウザーに応じて調整されます。

表 3. HATS スタイル・シート (続き)

名前	説明
smallFont	ブラウザー設定およびクラス関数に関連して生成されたすべての出力のフォント・サイズ
tantheme	黄褐色の背景。前景色はさまざまな色調のグレーと青です。
transport	Transport.jsp テンプレート用のスタイル・シート
whitetheme	白の背景。前景色はホスト画面の色と一致します。
xlargeFont	ブラウザー設定およびクラス関数に関連して生成されたすべての出力のフォント・サイズ
xsmallFont	ブラウザー設定およびクラス関数に関連して生成されたすべての出力のフォント・サイズ

「設計」タブの考慮事項と制限: 以下に、「設計」タブを使用する際の考慮事項および制限をリストします。

- テンプレートの正確なブラウザー・プレビューが行われるものとして「**設計**」タブを使用しないでください。代わりに、「**プレビュー**」タブを使用してください。
- 「**設計**」タブでは JavaScript は処理されません。そのため、JavaScript によってページの視覚操作が行われる場合、その効果は示されません。
- 一部の CSS ファイルは「**設計**」タブに正確に表示されません。

ソース

「**ソース**」タブには、テンプレートのすべての部分について JSP ファイル内の HTML タグと JSP タグが表示されます。Page Designerの別のタブで変更を行うと、ソース・ファイルに表示されるタグと属性もそれに合わせて変更されます。

ソース・ファイル内のタグや属性を変更することもできます。変更すると、Page Designerの他のタブに反映されます。

分割

「**分割 (Split)**」タブには 2 つのビューが表示されます。ツールバー・ボタンを使用して、「**設計**」ビューと「**ソース**」ビューの両方、または「**プレビュー**」ビューと「**ソース**」ビューの両方の表示を構成できます。

プレビュー

「**プレビュー**」タブでは、 <HATS:Component > (およびすべての <HATS:*xxxx*>) タグのレンダリングの際に静的な HTML コンテンツを表示するテンプレートのブラウザー・プレビューが提供されます。このタブは「**デザイン**」タブに似ていますが、このタブでは変更を行うことはできません。このプレビューには変換が含まれません。

注:

1. プレビュー設定は、ブラウザー全体に継承されます。プレビューのアクセス可能度は、ブラウザー全体で表示されるページのアクセス可能度に限定されます。
2. プレビューは、実行時にホストからのみ使用可能な項目では機能しません。

カスタマイズされたテンプレート、スタイル・シート、およびイメージの使用可能化

HATS では、カスタマイズされたテンプレート、スタイル・シート、およびイメージを共用できます。

- カスタマイズしたテンプレートを共用するには、以下からテンプレート .jsp ファイルをコピーします。

  <workspace_directory>¥project name¥Web Content¥templates

  これを以下にコピーします。

  <shared_install_directory>¥plugins¥com.ibm.hats_*nnn*¥predefined¥projects¥*language*¥new¥Web Content¥templates

  ここで、*shared_install_directory* は IBM Installation Manager を使用して HATS オファリングをインストールした共用リソース・ディレクトリーを、*nnn* は HATS のバージョンおよびビルド・レベルを表し、*language* は英語の場合は en です。

- Finance.jsp、Industry.jsp、Research.jsp、または Transport.jsp テンプレートのカスタマイズ版をコピーする場合は、以下から適切な JavaScript ファイルをコピーする必要もあります。

  <workspace_directory>¥project name¥Web Content¥common¥scripts

  to

  <shared_install_directory>¥plugins¥com.ibm.hats_*nnn*¥predefined¥projects¥new¥Web Content¥common¥scripts

- カスタマイズされたスタイル・シートをすべての新規 HATS プロジェクトで使用できるようにするには、以下から .css ファイルをコピーします。

  <workspace_directory>¥project name¥Web Content¥common¥stylesheets

  これを以下にコピーします。

  <shared_install_directory>¥plugins¥com.ibm.hats_*nnn*¥predefined¥projects¥new¥Web Content¥common¥stylesheets

- カスタマイズされたイメージをすべての新規 HATS プロジェクトで使用できるようにするには、以下から .jpg、.gif、または .png ファイルをコピーします。

  <workspace_directory>¥project name¥Web Content¥common¥images

  to

  <shared_install_directory>¥plugins¥com.ibm.hats_*nnn*¥predefined¥projects¥new¥Web Content¥common¥images

  注: 一部のテンプレート・イメージは、¥images サブディレクトリーのサブディレクトリーにあります。

これらのファイルをコピーすると、次に新規プロジェクトを作成するときに、新規のテンプレート、スタイル・シート、またはイメージを選択できます。

リッチ・クライアント・プロジェクトのテンプレートの編集

HATS リッチ・クライアント・テンプレートは、特定の Java インターフェースを実装している特殊な Java SWT コンポジット・クラスです。このインターフェースを使用すると、リッチ・クライアント環境で HATS テンプレートとして使用される

Java ソース・ファイルを HATS で検出して使用できます。HATS リッチ・クライアント・テンプレートは、単純なヘッダーまたはサイドバーのみで、フッターのない、きわめて基本的な構成になっています。この理由は、ランタイム環境 (例えば、Eclipse RCP、Lotus Notes、Lotus Expeditor など) は、全体的に共通の外観、および共通のボタンやリンクなどの領域を備えていることが前提となるためです。

テンプレートは、ホスト・コンポーネント (ComponentRendering コンポジット)、デフォルト・レンダリング・コンポジット、またはグローバル変数コントロールを含むことができませんが、変換はそれらを含むことができます。

すべてのテンプレート (HATS に付属のテンプレートと作成したテンプレート) を参照するには、「**HATS Projects**」ビューの「**リッチ・クライアント・コンテンツ**」の下にある「**テンプレート**」フォルダーを展開します。

デフォルト・エディターを使用してテンプレートを編集するには、テンプレートの名前をダブルクリックします。使用可能なエディターを参照するには、テンプレートの名前を右クリックして、「**オープン先を指定**」を選択します。

Java ビジュアル・エディター・フィーチャーが Rational SDP とともにインストールされていると想定すると、Visual Editor を使用してテンプレートを編集できます。このセクションの情報に加えて、「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」の章『テンプレート』を参照してください。また、ご使用の Rational SDP 製品の Knowledge Center も参照してください。例えば、http://publib.boulder.ibm.com/infocenter/radhelp/v8r5/index.jsp を参照し、`visual editor` を検索します。

Visual Editor の 2 つの領域は、通常は上段の領域として表示される「デザイン」ビューと、通常は下段の領域として表示される「ソース」ビューです。画像などの項目は、「**パレット**」ビューで使用できるコントロールを使用することにより、テンプレートに追加されます。

例えば、テンプレートに画像を追加するには、以下の手順に従います。

1. Rational SDP のメニュー・バーで、「**ファイル**」>「**インポート**」>「**一般**」>「**ファイル・システム**」を選択し、「インポート」ウィザードを開きます。
2. 「**ソース・ディレクトリー**」フィールドの横にある「**参照**」ボタンを使用して画像を検索して選択し、プロジェクトにインポートします。
3. 「**宛先フォルダー**」フィールドの横にある「**参照**」ボタンを使用して、プロジェクトの **¥src¥images** フォルダーを選択します。
4. 「**完了**」をクリックします。
5. 「HATS Projects」ビューで、プロジェクトを右クリックし、「**更新**」を選択します。
6. Visual Editor でテンプレートを開き、「**SWT コントロール**」の下にある「パレット」ビューで、「**ラベル**」をクリックします。
7. テンプレートの「デザイン」ビューで、画像を配置する場所にマウスのポイントを置いて**クリック**します。
8. 「名前」ダイアログで、名前を入力するか、または提供されているデフォルトをそのまま採用して「**OK**」をクリックします。

9. テンプレートに追加したばかりのラベルを選択し、通常は Visual Editor のソース・ビューの下に表示されている「プロパティー」ビューで、**text** プロパティーを選択して、ラベルの表題を消去します。
10. 「プロパティー」ビューで、**image** プロパティーを選択し、「**値**」列の右マージンに表示されている**ボタン**をクリックします。
11. 「イメージ・ファイルの選択」ダイアログで、「**参照対象**」の下で「**プロジェクト**」を選択し、「**ロケーション**」の下で **src** フォルダーを展開して **images** フォルダーをクリックし、「**ファイル**」の下で、インポートした画像を選択して「**OK**」をクリックします。

    注: このステップでは、「**参照対象**」の下で「**プロジェクト**」を選択してから、プロジェクト内のロケーションを選択するようにしてください。そのようにせず、「**ファイル・システム**」を選択してファイル・システムのロケーションを選択した場合、プロジェクトを別のシステムにデプロイしても、そのプロジェクトは正しく稼働しません。

Visual Editor を使用して、フォント、画像、色などのリソースを HATS の変換またはテンプレートに手動で追加する場合は、これらのリソースをクリーンアップするコードも手動で追加する必要があります。詳しくは、103 ページの『HATS リッチ・クライアントの考慮事項と制限』を参照してください。

アプリケーション・キーパッド

アプリケーション・キーパッドには、アプリケーション・レベルの機能を表すキー (「**最新表示**」、「**デフォルト**」、「**印刷ジョブを表示**」など) が含まれています。これらのキーは、HATS アプリケーション内の機能を制御します。アプリケーション・キーパッドのキーには以下のものがあります。

キーボードのオン/オフ
ホスト・キーボード上の物理キーを使用するかどうかを切り替えます。このプロジェクトについてキーボード・サポートが使用可能にされていない場合は、アプリケーション・キーパッドにはこのボタンは表示されません。詳しくは、423 ページの『第 16 章 キーボード・サポートの使用可能化』を参照してください。

リセット
ブラウザー・ページ上のすべてのフィールドから、ユーザーが行った入力をクリアします。

切断 ホスト・セッションから切断します。このキーをクリックすると、ユーザーがホストに再接続できるようにするためのリンクが表示されます。

画面を反転 (適用可能な場合)
アラビア語またはヘブライ語のコード・ページを使用するホストでアプリケーションを実行する場合に、画面イメージを、左から右へのイメージから右から左へのイメージに (またはその逆に) 切り替えます。プロジェクト接続設定でアラビア語またはヘブライ語のコード・ページが選択されていない場合は、アプリケーション・キーパッドにはこのボタンは表示されません。

デフォルト
変換によってホスト画面のカスタマイズをオフにし、ホスト画面全体を基本

HTML で表示します。デフォルト・キーを変更するには、変換において異なるテンプレートまたはスタイル・シートを選択します。

最新表示
現在のブラウザー画面を最新表示し、現在のアクションを再度実行します。

印刷ジョブを表示 (適用可能な場合)
ユーザーが作成した印刷ジョブのリストを表示します。このプロジェクトについて印刷サポートが使用可能にされていない場合は、アプリケーション・キーパッドにはこのボタンは表示されません。

デフォルトでは、HATS はアプリケーション・キーパッドを表示します。アプリケーション・キーパッドを非表示にするには、「**HATS Projects**」ビューに移動して HATS プロジェクトの「**プロジェクト設定**」をダブルクリックし、「**レンダリング**」タブを選択して「**アプリケーション・キーパッド**」をクリックします。 次に、チェック・ボックスをクリアして、キーパッドを非表示にします。

注: カスタム・キーパッドおよび個々のキーは、個々のタグまたはコンポジットを使用して定義するため、プロジェクト設定でキーパッドを使用不可にしている場合でもテンプレートのエディターには表示されます。

キーボード・サポートおよびキーパッドの設定の詳細については、 107 ページの『第 5 章 HATS プロジェクトの変更』を参照してください。

第 11 章 マクロとホスト端末

画面のセットを定義する XML スクリプトをマクロといいます。各画面には、画面の説明、画面に対して実行するアクション、およびアクションが実行された後で表示される 1 つまたは複数の画面セットが含まれる。

HATS は、ホスト・アプリケーション・プロセスの速度およびカスタマイズを高めるためのマクロ・ベースのカスタマイズをサポートしています。マクロは、次の機能で使用できます。

- マクロ実行アクションやマクロ・トランザクション実行アクションなどのアクションとして。マクロの実行は、画面イベントとして実行される最後のアクションでなければなりません。詳しくは、 181 ページの『アクション』を参照してください。
- 特定のホスト接続を接続または切断するため。詳しくは、 169 ページの『マクロ』を参照してください。
- 統合オブジェクトを生成するため。詳しくは、 404 ページの『統合オブジェクトの作成』を参照してください。
- 変換で使用する。詳細については、 217 ページの『マクロ・キーを挿入』を参照してください。

マクロを使用して、次の機能のいずれかまたはすべてを実行できます。

画面スキップ
画面スキップ・マクロは、途中の画面を表示せずに、1 つの画面から別の画面へとユーザーを導くナビゲーション・マクロです。

プロンプト
プロンプト・マクロには、ホスト・セッション中にユーザーに入力を要求するステップが含まれています。ユーザー・リストからプロンプトを設定することもできます。例えば、プロンプト・マクロを使用して、ホスト・アプリケーションにログインする前に、ユーザーにユーザー ID とパスワードの入力を要求することができます。

注: ユーザー・リスト・プロンプトは、接続マクロでのみ使用できます。

プロンプト・マクロは、開発者が定数の値、またはグローバル変数から取得した値を提供するために使用することもできます。

抽出 抽出マクロには、ホスト画面情報をストリング、ストリングのリスト、またはテーブルとして抽出するイベントが含まれています。例えば、抽出マクロを使用して、ディレクトリー・タイプのホスト・アプリケーションに接続し、そのディレクトリー内での検索の結果を抽出することができます。

HATS Toolkit に記録またはインポートされているマクロは、HATS マクロ・ファイル (.hma) に保存されます。プロジェクト内で定義されているマクロを見るには、HATS Toolkit の 「**HATS Projects**」ビュー内の 「**マクロ**」ノードを展開します。

© Copyright IBM Corp. 2003, 2015

HATS には、HATS マクロの表示と変更に使用できる 3 つの異なるエディターが用意されています。マクロ・エディター、拡張マクロ・エディター、および Visual Macro Editor です。 Visual Macro Editor が HATS マクロのデフォルト・エディターです。マクロ名をダブルクリックすると、Visual Macro Editor が開きます。Visual Macro Editor の詳細については、*「HATS 拡張マクロ・ガイド」*の『Visual Macro Editor』の章を参照してください。

マクロ・エディターを開くには、マクロの名前を右クリックして、**「アプリケーションから開く」>「マクロ・エディター」**を選択します。

拡張マクロ・エディターは、マクロ・エディターの「概要」ページから開きます。拡張マクロ・エディターの詳細については、*「HATS 拡張マクロ・ガイド」*の『拡張マクロ・エディター』の章を参照してください。

注: Macro Editor および Advanced Macro Editor のサポートは、HATS V9.5 では非推奨になりました。現在はサポートを継続していますが、IBM は、これ以降の製品リリースでこれらの機能を削除する権利を有しています。このサポートは Visual Macro Editor に置き換えられます。

マクロ・エディター

以下のセクションでは、マクロ・エディターの各タブについて説明します。

概要

マクロ・エディターの**「概要」**タブには、名前、説明、最終変更日時など、マクロに関する一般情報が表示されます。このタブで変更できる項目は、マクロの説明、一時停止時間、タイムアウト、および接続です。

「一時停止時間」は、各アクションの実行後の遅延のミリ秒数です。**「タイムアウト」**は、マクロを終了するまでに待つミリ秒数です。**「接続」**では、使用する接続をドロップダウン・メニューから選択できます。

「ホスト端末をオープン」ボタンと**「拡張エディター」**ボタンについては、以下で説明します。

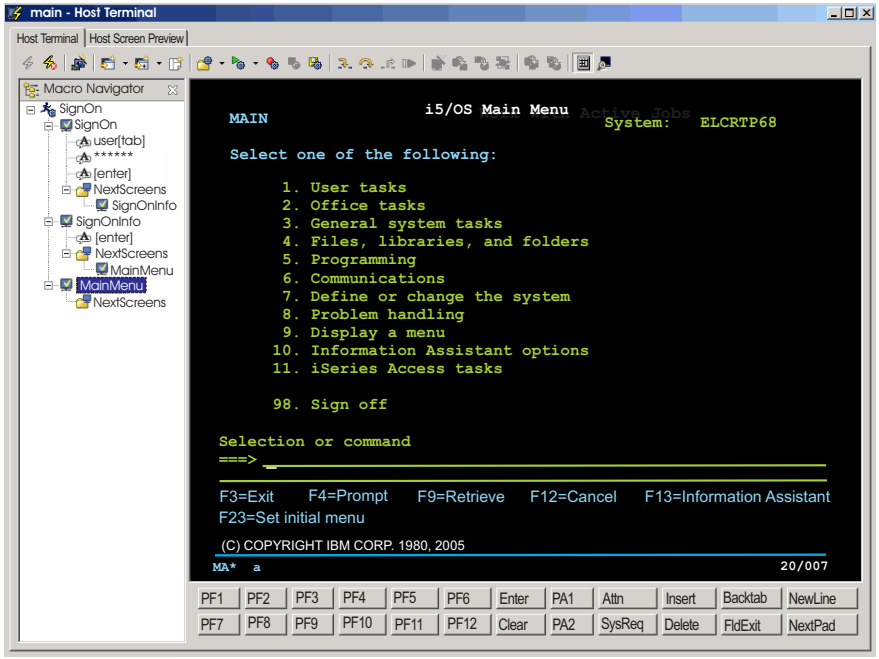


図 75. HATS マクロの記録

ホスト端末をオープン

ホスト端末を使用してマクロを作成または処理するには、「**ホスト端末をオープン**」をクリックします。端末が開いたら、ツールバー上のアイコンを使用してマクロを記録および作成します。

ホスト端末: 「**ホスト端末**」を使用して、マクロ・ツリー内のノードを右クリックし、「**編集**」を選択することによって、前に記録したマクロを編集することもできます。編集する画面を選択すると、画面認識基準エディターが起動し、画面を変更できます。画面に入力したデータを編集する場合は、起動したウィンドウで入力データを変更します。「**次の画面**」ノードを編集すると、次の画面として選択可能な画面のリストが表示されます。リストから、画面に表示する順に選択します。

「**ホスト端末**」ツールバーには、マクロの実行エラーをデバッグするためのアイコンが 4 つあります。アイコンはマクロをステップごとに実行できるよう設計されています。マクロでは、ステップが 1 つのアクションを実行したり 1 つの画面を認識したりするように定義されています。アイコンは、「**ステップイントゥ**」、「**ステップオーバー**」、「**ステップ・リターン**」、および「**再開**」です。

各アイコンには、ファンクション・キーが関連付けられています。そのときアクティブなビュー (「**マクロ・ナビゲーター**」または「**ホスト端末**」のいずれか) によって、F5、F6、F7、および F8 キーの機能が異なります。例えば、「**マクロ・ナビゲーター**」がアクティブである場合、F5 を押すと「**ステップイントゥ**」、F6 を押すと「**ステップオーバー**」となります。「**ホスト端末**」がアクティブである場合、F5 または F6 を押すと、ホストに AID キーが送信されます。上記およびその他のマクロ関連アイコンについて詳しくは、マクロ関連アイコンを参照してください。

注:

1. HATS では、一定の画面に画面認識基準を定義する必要があります。対象となる画面を以下に示します。

- 最初および最後のマクロ画面
- ループの開始および終了
- マクロ抽出
- マクロ・プロンプト

2. ホスト端末では、以下のキーの組み合わせを使用することにより、切り取り、コピー、および貼り付け機能が提供されます。
   - 切り取り: Ctrl-X または Shift-Delete
   - コピー: Ctrl-C または Ctrl-Insert
   - 貼り付け: Ctrl-V または Shift-Insert
3. 2 つの新規ビデオ端末 (VT) 接続プロパティーを使用して、VT ホスト端末の通常の前景または背景を、選択した色に強制設定できます。これは、VT アプリケーションが背景色に合うように前景色を変更したためにホスト端末上のテキストが判読できなくなった場合に役立つと考えられます。

   接続エディターの「**拡張**」タブに設定を追加できます。設定には **forceVTNormalForegroundToColor** と **forceVTNormalBackgroundToColor** があり、これらは相互に独立して作動します。

   設定値の形式は `redValue,greenValue,blueValue` であり、それぞれの値の範囲は 0 から 255 です。例えば、通常の前景色を白色にオーバーライドするには、次の設定を拡張設定のテーブルに追加します。

   ```
   forceVTNormalForegroundToColor = 255,255,255
   ```

   これらの設定は、通常の前景色と背景色にのみ作用します。文字属性 (太字、明滅、下線、リバースなど) が設定されている場合、または ANSI カラーが VT アプリケーションで設定されている場合、これらの設定は無視されます。

   拡張接続設定について詳しくは、 161 ページのオプションの拡張接続設定を構成してくださいを参照してください。

表 4 に、ホスト端末画面ツールバーに表示されるマクロ・アイコンを示します。

表 4. マクロ・アイコン

アイコン	説明
	マクロをオープンします。ドロップダウン・リストからマクロを選択します。
	マクロを再生します。ドロップダウン・リストからマクロを選択します。
	マクロの記録を開始します。
	マクロの記録を停止します。
	記録されたマクロを保存します。
	デバッグ目的でマクロにステップイントゥします (F5)。

表 4. マクロ・アイコン (続き)

アイコン	説明
	デバッグ目的でマクロをステップオーバーします (F6)。
	デバッグ目的でマクロをステップ・リターンします (F7)。
	マクロの実行を再開します (F8)。
	マクロ画面の画面認識基準を定義します。
	現在のマクロにプロンプト・アクションを追加します (マクロの記録時にのみ使用可能)。
	現在のマクロに抽出アクションを追加します (マクロの記録時にのみ使用可能)。
	現在のマクロにループを記録します (マクロの記録時にのみ使用可能)。
	画面上のすべてのフィールドのため、現在のマクロにプロンプト・アクションを追加します (マクロの記録時にのみ使用可能)。
	画面上のすべてのフィールドのため、現在のマクロに抽出アクションを追加します (マクロの記録時にのみ使用可能)。
	ホスト端末ウィンドウのキーパッドを表示または非表示にします。
	テキスト・ベースのオペレーター情報域 (OIA) を表示または非表示にします。

マクロ関連アイコン:

マクロの記録

HATS ホスト端末を使用して、HATS Toolkit にマクロを記録することができます。そのためには、HATS ホスト端末画面で、「**マクロの記録**」アイコンをクリックします。「マクロの記録」ウィザードが表示され、ターゲット・プロジェクトの選択、マクロの名前の指定、マクロの説明の入力、およびマクロの保存場所の確認を行うことができます。「**記録中に自動的に画面キャプチャーを作成**」オプションにより、マクロの記録中にナビゲートする各画面の画面キャプチャーを自動作成するよう指定できます。こうして作成されたこれらの画面キャプチャーは、後でマクロに変更を加える必要があるときに、 Visual Macro Editor で使用できます。このオプションはデフォルトで選択されています。これらの項目を指定したら、「**完了**」をクリックします。

次に、「**画面認識基準の定義**」ウィンドウが表示されます。HATS では、最初の画面に、明示的に定義された画面認識がなくてはならないためです。これで、HATS ホスト端末画面を使用して、ホスト・アプリケーション内の任意の画面にナビゲートできるようになります。

マクロが HATS ホスト端末内で既に開いている場合、「**マクロの記録**」アイコンが、新規マクロを記録するか開いているマクロに追加するかを尋ねます。開いているマクロに追加すると、マクロの記録が開始され、現在のマク

ロにアクションが追加されます。記録を開始するマクロ内の位置は、左側の「**マクロ・ナビゲーター**」ツリーで選択された内容と、選択されたノードの画面認識基準に現在の端末画面が一致するかどうかによって異なります。具体的には、次のとおりです。

- 「**マクロ・ナビゲーター**」で画面ノードまたはアクション・ノードが選択されていて、選択された画面の認識基準と現在のホスト端末画面が一致する場合、新規のアクションは選択されたノードの先頭に追加されます。一致する画面ノード内でアクションが選択された場合、一致する画面内で選択されたアクションの直後に記録が開始されます。
- 「**マクロ・ナビゲーター**」ウィンドウで画面ノードまたはアクション・ノードが選択されていて、ホスト端末画面が選択した画面の認識基準と一致しない場合、選択したノードの「**次の画面**」に新規画面ノードが追加され、新規画面ノードで記録が開始されます。
- マクロ名に対応する「**マクロ・ナビゲーター**」ウィンドウで上位ノードが選択されている場合、HATS はすべての画面ノードを検索し、現在の端末画面と一致するものを見つけ出そうとします。一致するものがあった場合、一致する画面の先頭から記録が開始されます。一致するものがなかった場合、最後の画面の「**次の画面**」に新規画面ノードが追加され、新規画面ノードで記録が開始されます。

抽出を挿入するステップまたはプロンプトを追加するステップのみのマクロは作成しないでください。作成すると、ユーザーが値を指定したときにページが変更されず、ユーザーが別のページへ移動することができなくなります。マクロがホストに戻らない場合、タイムアウトになるまで滞留します。

注: マクロから統合オブジェクトを作成する場合、以下の要件に従ってマクロを命名してください。

- マクロの名前には、英字、数字、または下線文字のみを使用できます。
- 名前の最初の文字を数字にすることはできません。
- 2 バイト (全角) 文字は使用できません。

マクロの停止

マクロの記録を終了したら、「**マクロを停止**」アイコンをクリックします。

マクロの保存

マクロの記録を停止したら、「**マクロを保存**」アイコンをクリックしてマクロを保存する必要があります。

ステップイントゥ (Step Into) (F5)

「**ステップイントゥ (Step Into) (F5)**」アイコンをクリックすると、画面から、組み込まれた最初のアクションに入り、アクションを 1 つずつ実行できます。キーボードで F5 キーを使用しても、同じ機能が実行できます。これは、デバッグの目的で使用します。

ステップオーバー (Step Over) (F6)

「**ステップオーバー (Step Over) (F6)**」をクリックすると、画面から画面に (間にあるアクションをすべて実行しながら) 移動できます。キーボードで F6 キーを使用しても、同じ機能が実行できます。これは、デバッグの目的で使用します。

ステップ・リターン (Step Return) (F7)

「**ステップ・リターン (Step Return) (F7)**」をクリックすると、画面の下のアクションの実行を終了し、次の画面に移動します。キーボードで F7 キーを使用しても、同じ機能が実行できます。これは、デバッグの目的で使用します。

再開 (Resume) (F8)

マクロの実行を再開するには、「**再開 (F8)**」アイコンをクリックするか、キーボードで F8 キーを押します。現在、マクロを実行している場合、これによって残りのマクロをすべて実行できます。

画面認識基準の定義

HATS がホスト画面の突き合わせに使用する画面認識基準を設定できます。画面上の入力フィールド数または合計フィールド数、カーソル位置の座標、および、画面上に定義された長方形枠内または画面の任意の場所にあるテキスト・ストリングなどの基準を任意に組み合わせて、ホスト画面を認識させることができます。詳しくは、175 ページの『画面認識基準または開始画面』を参照してください。

プロンプト・アクションを追加

ユーザーに情報の入力を求めるマクロが必要な場合は、「**プロンプト・アクションを追加**」をクリックして「プロンプト・アクションを追加」ウィザードを表示します。ここで、プロンプトの名前とデフォルト値を指定できます。パスワードのようなユーザーが入力する情報が、ホスト画面に表示してはならない情報である場合は、「**パスワード非表示**」チェック・ボックスをクリックします。

注: プロンプトに対して指定するデフォルト値は、暗号化されずにマクロ・ファイルに保管されます。マクロ・エディターを使用してプロンプトを編集する場合、デフォルト値は平文で表示されます。したがって、プロンプトを使用してパスワードを指定することは適切ですが、セキュリティー上の理由から、パスワードのデフォルト値は指定しないでください。

このウィザードの「**位置**」セクションの「**行**」および「**列**」フィールドは、ユーザーが入力したプロンプト情報をホスト画面上のどこに配置するかを定義します。マクロの記録を開始する前に、ホスト画面上の特定の場所 (例えばパスワード用のフィールド) にカーソルを置いた場合は、「**行**」および「**列**」フィールドにはそのカーソル位置の値が入っています。ウィザードの「**マクロ・プロンプトを処理**」セクションでは、プロンプトをどのように処理するかを指定できます。次のいずれかのラジオ・ボタンを選択できます。

ハンドラーを表示

HATS Web プロジェクトの場合、ユーザーに必要な情報の入力を求めるための .jsp ファイルを選択し、ユーザーが情報を送信するためのボタンを組み込むことができます。HATS には default.jsp という名前のデフォルトのマクロ・ハンドラーが付属しています。このファイルを見つけるには、HATS Toolkit の「**HATS Projects**」ビューをクリックし、プロジェクト名を展開して、「**Web コンテンツ**」>

「**マクロ・イベント・ハンドラー**」の順に展開します。独自のハンドラーを作成する場合は、必ず HATS ランタイムに制御を戻してください。

注:

1. 統合オブジェクトでは、選択された .jsp ハンドラーを無視します。代わりに、統合オブジェクト用の入力ページが作成され、値の入力を求めるプロンプトが入力ページ内に配置されます。生成された出力ページでは、統合オブジェクトが実行される前に、入力ページから渡された値を統合オブジェクトにコピーします。
2. マクロがテキスト平面以外の平面から抽出する場合、「**ハンドラーを表示**」オプションは使用不可です。

注:

HATS リッチ・クライアント・プロジェクトの場合、ユーザーに必要な情報の入力を求めるために、カスタム・マクロ・ハンドラーを指定するか、リッチ・クライアント・プロジェクトに定義されたカスタム・マクロ・ハンドラーのリストを参照して選択することができ、さらにユーザーが情報を送信するためのボタンを組み込むことができます。DefaultMacroHandler という名前のデフォルトのマクロ・ハンドラーが HATS とともに出荷されます。「**HATS Projects**」ビューでこのファイルを見つけるには、プロジェクト名を展開して、「**リッチ・クライアント・コンテンツ**」>「**マクロ・イベント・ハンドラー**」の順に展開します。独自のハンドラーを作成する場合は、ご使用のカスタム・マクロ・ハンドラーの render() メソッド内の createButtonArea() と createButtons() メソッドを呼び出して、必ず HATS ランタイムに制御を戻してください。これらのメソッドは、RcpMacroHandler.render() メソッドのデフォルトのインプリメンテーションで呼び出されます。RcpMacroHandler について詳しくは、HATS Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API 参照』セクションを参照してください。

プロンプトをストリングに設定

プロンプトから戻す値がわかっている場合は、そのストリングを「**ストリング**」フィールドに入力できます。

プロンプトをグローバル変数に設定

プロンプトの値をグローバル変数により入力する場合は、そのグローバル変数の名前を「**名前**」フィールドに入力するか、「**グローバル変数**」フィールドの隣のドロップダウン・メニューを使用して既存の変数を選択します。「**拡張**」ボタンをクリックすると、変数が共用または索引付きであるかどうかを指定できます。索引付き変数の場合は、すべての索引を表示するのか、単一の索引を表示するのかを指定する必要があります。詳しくは、399 ページの『第 12 章 グローバル変数との対話』を参照してください。

注:

プロンプトの値が、抽出によって設定されたグローバル変数に基づいていて、**promptall** が `true` に設定されている場合、プロンプトの値が取り出されるまで抽出アクションは実行されません。このため、プロンプトによって使用されるグローバル変数には値が入っていません。抽出とプロンプトにグローバル変数を使用する場合は、**promptall** を `false` に設定する必要があります。

統合オブジェクトではグローバル変数に直接アクセスしません。代わりに、統合オブジェクト用の入力ページおよび出力ページはグローバル変数の値を取得し、実行前に統合オブジェクトに設定します。統合オブジェクトからアクセスできるのは、共用グローバル変数のみです。

プロンプトをユーザー・リストからのプロパティーに設定

プロンプトでユーザー・リストにアクセスするには、ドロップダウン・リストから「**ユーザー・プロファイル**」を選択します。ユーザー・プロファイルは、「**ユーザー・リスト・プロパティー**」として、**ユーザー** `ID` または パスワードを使用するかどうかの決定に関するキーになります。詳しくは、171 ページの『ユーザー・リスト』を参照してください。

注: ユーザー・リスト・プロンプトは、接続マクロでのみ使用できます。

Web 高速ログオンを使用する

Web 高速ログオンを使用するように HATS アプリケーションを構成してある場合は、「**プロンプト・タイプ**」ドロップダウン・リストでプロンプト・タイプをユーザー ID またはパスワードとして入力し、「**アプリケーション ID**」フィールドにアプリケーション ID を入力します。

注: 「**プロンプト・タイプ**」には正しい値が事前に入力されている必要があります。

選択を終えたら「**OK**」をクリックします。

注: 双方向言語サポートを使用する場合の考慮事項については、526 ページの『マクロのプロンプトおよび抽出の双方向オプション』を参照してください。

抽出アクションを追加

ホスト画面から情報を抽出するマクロが必要な場合は、ホスト画面上の領域を選択し、「**抽出アクションを追加**」をクリックして「抽出アクションを追加」ウィザードを表示します。 ここでは抽出の名前を指定できます。このウィザードの「**位置**」セクションの「**開始行**」、「**開始列**」、「**終了行**」、および「**終了列**」フィールドは、ホスト画面上のどこから情報を抽出するかを定義します。長方形でホスト画面上の領域を指定してから「**抽出アクションを追加**」をクリックすると、「**位置**」セクションのフィールドに、「抽出アクションを追加」ウィザードが表示されたときの値が入ります。

「**抽出形式**」は、以下のいずれかに設定できます。

- **この領域を 1 つのストリングとして抽出**

  抽出されたテキストを 1 行の文字ストリングとして保存することを指定します。このオプションは、統合オブジェクト、グローバル変数、およびマクロ・ハンドラーに対してサポートされます。
- **この領域をストリングのリストとして抽出**

  抽出されたテキストを複数行のストリングのリストとして保存することを指定します。このオプションは、グローバル変数およびマクロ・ハンドラーに対してサポートされます。統合オブジェクトでは、ストリングのリストは 1 つの連続するストリングとして扱われます。
- **この領域をテーブルとして抽出**

  抽出されたテキストを、縦横に広がりを持つ (行と列がある) ストリングのテーブルとして保存することを指定します。このオプションは、統合オブジェクトおよびマクロ・ハンドラーに対してサポートされます。グローバル変数の場合、テーブルとして抽出された文字列は、1 つの連続した文字列になります。

  テーブル抽出の構成

  このページを使用してテーブルの列の形式を設定します。

  – **列名**

    このフィールドを使用して、選択した列の見出しを変更します。

  – **列の展開**

    列の展開は、文字を列間で移動させます。ある列を強調表示した状態にします。「**左方**」ボタンにより、その列の各行の最後の文字が強調表示した列の左側に移動し、強調表示した列の各行の最初の文字になります。「**右**」ボタンは、強調表示した列の右側にある列の各行から先頭の 1 文字を移動して、強調表示した列の各行の最終文字にします。

  – **列の削減**

    列の削減は、文字を列間で移動させます。ある列を強調表示した状態にします。「**左方**」ボタンにより、強調表示した列の各行の最初の文字が移動して、強調表示した列の左隣の列の各行の最後の文字になります。「**右**」ボタンは、強調表示した列の各行から最終の 1 文字を移動して、強調表示した列の右側にある列の各行の先頭文字にします。

  – **マージ**

    このボタンは、2 つの強調表示した列を単一の列にマージするために使用します。強調表示された列の文字は、1 列に結合されます。

  – **分割**

このボタンは、強調表示した 1 つの列を 2 つの別々の列に分割するために使用します。強調表示した列の文字列は 2 つの新規の列間で等しく分割されます。文字数が奇数の場合、左の列が右の列よりも 1 文字多くなります。

注:

DBCS サポートとともにこの機能を使用する場合の考慮事項については、549 ページの『マクロの記録』を参照してください。

ウィザードの「**マクロ抽出を処理**」セクションでは、プロンプトをどのように処理するかを指定できます。次のチェック・ボックスを選択できます。

ハンドラーを表示

HATS Web アプリケーションの場合、抽出された情報をユーザーに表示するための .jsp ファイルを選択できます。HATS には、default.jsp という名前のデフォルトのマクロ・ハンドラーが組み込まれています。「**HATS Projects**」ビューでこのファイルを見つけるには、プロジェクト名を展開して、「**Web コンテンツ**」>「**マクロ・イベント・ハンドラー**」の順に展開します。独自のハンドラーを作成する場合は、必ず HATS ランタイムに制御を戻してください。

注: 統合オブジェクトではこのオプションを使用しません。出力ページでは、代わりに、統合オブジェクトから抽出されたデータを取得して表示します。

HATS リッチ・クライアント・プロジェクトの場合、ユーザーに必要な情報の入力を求めるために、カスタム・マクロ・ハンドラーを指定するか、リッチ・クライアント・プロジェクトに定義されたカスタム・マクロ・ハンドラーのリストを参照して選択することができ、さらにユーザーが情報を送信するためのボタンを組み込むことができます。HATS では、DefaultMacroHandler という名前のデフォルトのマクロ・ハンドラーが提供されています。「**HATS Projects**」ビューでこのファイルを見つけるには、プロジェクト名を展開して、「**リッチ・コンテンツ (Rich Content)**」>「**マクロ・イベント・ハンドラー**」の順に展開します。独自のハンドラーを作成する場合は、ご使用のカスタム・マクロ・ハンドラーの render() メソッド内の createButtonArea() と createButtons() メソッドを呼び出して、必ず HATS ランタイムに制御を戻してください。これらのメソッドは、RcpMacroHandler.render() メソッドのデフォルトのインプリメンテーションで呼び出されます。RcpMacroHandler について詳しくは、HATS Knowledge Center (http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0?topic=/com.ibm.hats.doc/doc/rcpjavadoc/index.html) の『HATS RCP API 参照』セクションを参照してください。

グローバル変数として保存

グローバル変数の名前は「**名前**」フィールドに入力できます。または、ドロップダウン・メニューを使用して既存の変数を選択するこ

ともできます。「**名前**」フィールドで既存のグローバル変数を選択した場合は、次のいずれかのラジオ・ボタンを選択して、その既存の変数をどのように処理するのかを指定する必要があります。

- この新しい値で既存の値を上書き
- 特定の索引から開始して既存の値をこの新しい値で上書き。
- この新しい値を既存の値の末尾に追加。 (Append this new value to the end of the existing value.)
- 特定の索引でこの新しい値を既存の値に挿入。

「**共用**」チェック・ボックスを選択することにより、この変数を共用するかどうかを指定することもできます。

注: 値を抽出し、抽出によって設定されたグローバル変数にその値を割り当て、グローバル変数の値をプロンプトに使用する場合は、**promptall** を `false` に設定する必要があります。**promptall** が `true` に設定されていると、プロンプトの値が取り出されるまで抽出アクションは実行されません。このため、プロンプトで使用されるグローバル変数には値がありません。

統合オブジェクトではグローバル変数に直接抽出しません。代わりに、統合オブジェクトの出力ページでは、実行後の統合オブジェクトからデータを取得して、グローバル変数を設定します。統合オブジェクトからアクセスできるのは、共用グローバル変数のみです。

選択を終えたら「**OK**」をクリックします。

注: 双方向言語サポートを使用する場合の考慮事項については、 526 ページの『マクロのプロンプトおよび抽出の双方向オプション』を参照してください。

ループの記録

マクロ内でループを記録する場合は、「**ループの記録**」をクリックしてから、ループが開始された画面に移動します。「**次へ**」をクリックし、最初の画面の画面認識基準を定義します。「**次へ**」をクリックして追加の基準を定義するか、「**完了**」をクリックしてメインの端末に戻ります。メインの端末で、ループの各サイクルで実行されるアクションを実行し、「**次へ**」を押します。「**固有の画面が認識されたときに終了します**」 または「**一定の反復回数の後に終了します**」を選択してループの終了方法を決定し、後者の場合はその下にループの数を入力します。「**次へ**」をクリックしてループの最後の画面へ移動し、「**完了**」を押してループを終了します。

注: ループの画面を開始および終了するための画面認識基準は非常に重要です。開始画面と終了画面がほとんど同じ場合もあります。これらの画面を 1 つずつ注意深く区別する必要があります。

すべてのフィールドにプロンプト・アクションを追加

このアイコンを選択すると、「**グローバル変数からすべてのフィールドを挿入**」 ウィンドウが開きます。そこで、ドロップダウン・リストから「**格納済み画面名**」を選択します。「**画面キャプチャーを作成**」 チェック・ボッ

クスをオンにして画面キャプチャーを作成することもできます。「**共用**」チェック・ボックスを選択すると、共用リストからグローバル変数が取得されます。

注:

1. プロンプトに対して指定するデフォルト値は、暗号化されずにマクロ・ファイルに保管されます。ホスト端末のマクロ・エディターまたは拡張マクロ・エディターでプロンプトを編集する場合、デフォルト値は平文で表示されます。したがって、プロンプトを使用してパスワードを指定することは適切ですが、セキュリティー上の理由から、パスワードのデフォルト値は指定しないでください。
2. 統合オブジェクトにこのオプションを使用すると、HATS によって、定義されたグローバル変数それぞれの統合オブジェクトにフィールドが追加されます。グローバル変数の値を設定するコードを記述し、実行前の統合オブジェクトにコピーする必要があります。これは、統合オブジェクトを実行する .jsp ページで実行できます。詳しくは、413 ページの『統合オブジェクト・プロパティーの挿入』を参照してください。統合オブジェクトからアクセスできるのは、共用グローバル変数のみです。

すべてのフィールドに抽出アクションを追加

このアイコンを選択すると、「**グローバル変数からすべてのフィールドを抽出 (Extract All Fields from Global Variables)**」ウィンドウが開きます。そこで、ドロップダウン・リストから「**格納済み画面名**」を選択します。これは、すべてのフィールド用に作成されるグローバル変数の接頭部として使用されます。保護フィールドのテキストを実行時に取得する場合は、「**全フィールド**」オプションを選択します。保護フィールドのテキストを変換 JSP 内で静的にする場合は、「**入力フィールド**」オプションを選択します。「**共用**」チェック・ボックスを選択すると、共用リストからグローバル変数が取得されます。

注: 統合オブジェクトにこのオプションを使用すると、HATS によって、定義されたグローバル変数それぞれの統合オブジェクトにフィールドが追加されます。実行後の統合オブジェクトから共用グローバル変数にデータをコピーするコードを記述する必要があります。これは、統合オブジェクトを実行する .jsp ページで実行できます。詳しくは、413 ページの『統合オブジェクト・プロパティーの挿入』を参照してください。統合オブジェクトからアクセスできるのは、共用グローバル変数のみです。

キーパッドの表示/非表示

このアイコンでは、端末キーパッドを表示または非表示にすることができます。

テキスト OIA の表示

「**テキスト OIA の表示**」 (オペレーター情報域) アイコンを使用すると、端末画面上のカーソルの正確な行と列の位置が示されます。また、入力が禁止されているかいないかも示されます。

ホスト画面のプレビュー: サーバーによる変換後、現在のホスト画面がどのような外観になるかを確認するには、「ホスト端末」ウィンドウの上部にある「**ホスト画**

面のプレビュー (Host Screen Preview)」タブをクリックします。プレビュー画面では、現在のホスト画面を Web ページとして表示します。

ホスト画面のプレビューでは、最初に、現在の画面に定義された画面のすべてのカスタマイズを適用します。現在の画面に画面カスタマイズが定義されていない場合、プロジェクト・レベルのテンプレートとデフォルトのレンダリング規則が画面に適用されます。

ホスト画面のプレビューでは、「**プロジェクト設定**」の「**イベント**」タブにリストされた画面カスタマイズの順序を適用します。一致する画面認識を検出すると、即座に画面カスタマイズに定義された変換によって変換された画面を表示します。一致するものが見つからない場合、デフォルトの変換が表示されます。

ホスト画面のプレビューは、次の画面のリストに定義された画面のカスタマイズを処理しません。次の画面はホスト画面プレビューで処理されません。詳しくは、197 ページの『次の画面』を参照してください。

注: 「**ホスト画面のプレビュー**」タブは、現在のホスト端末がデフォルト接続に関連付けられている場合にのみ表示されます。バックグラウンド接続でホスト端末を開くと、このプレビューは表示されません。

拡張エディター

「**概要**」タブの「**拡張エディター**」をクリックすると、HATS マクロ・エディターの拡張機能を起動し、マクロの設定を変更することができます。マクロ・エディター専用のウィンドウが開き、次のタブが表示されます。

- マクロ
- 画面
- リンク
- 変数

マクロ・エディターの拡張エディター機能の詳細については、「*HATS 拡張マクロ・ガイド*」を参照してください。

プロンプトおよび抽出

マクロ・エディターの「**プロンプトおよび抽出**」タブには、構成されているマクロ・プロンプトと抽出のリストと、マクロの実行時にそれらがどのように処理されるかが示されます。テーブル内のプロンプトまたは抽出のいずれかを選択し「**編集**」をクリックすると、そのプロンプトまたは抽出の HATS 固有のプロパティーを編集することができます。

ソース

「**ソース**」タブには、macro-name.hma ファイルでマクロのすべての属性と値に関して使用されるタグが表示されます。macro-name は、マクロの作成またはインポート時にマクロに付けた名前です。プロジェクト・エディターの他のタブで変更を行うと、ソース・ファイルに表示されるタグと属性もそれに合わせて変更されます。ソース・ファイル内でタグおよび属性を変更することもでき、その場合は、マクロ・エディターの該当タブに変更が反映されます。

プロンプトおよび抽出は、記録された順番でマクロ・ソース・ファイルに表示されます。また、エディターでマクロを開くと、カーソルは、マクロ・ファイルの一番上に表示されるのではなく、最後にマクロを保存したときに置かれていた行と同じ行に表示されます。

デフォルトでは、コンテンツ・アシスタンスは、プロジェクト内のすべてのマクロで使用可能になっています。「**ソース**」タブでマクロを編集中にコンテンツ・アシスタンスを起動するには、Crtl+Space を押してください。製品内のどのマクロでコンテンツ・アシスタンスを提供するかを構成することができます。手順については、147 ページの『マクロ・コンテンツ・アシスタンス』を参照してください。

マクロ・エラーの処理

マクロのエラーは、マクロ・エディター、テキスト・エディター、または XML エディターのいずれを使用している場合にも、「**問題**」 ビューに要約されます。マクロのソース・コードは、エラーのあるすべての行に、赤い円の中に白の X が付いたマーカーを表示します。

マクロはエラーがある場合でも保存することができ、ユーザーは後で必要に応じてエラーを見つけて修正することができます。ただし、エラーのあるマクロは、実行時またはホスト端末で実行することはできません。

マクロのインポート

他のプログラム (IBM WebSphere Host Publisher や IBM Host On-Demand など) で記録されているマクロをインポートすることもできます。このようなマクロをインポートするには、「**ファイル**」>「**インポート**」>「**HATS**」>「**Host Publisher/HOD マクロ**」を選択し、「**次へ**」をクリックして、「**Host Publisher/HOD マクロをインポート**」ダイアログを表示します。「**追加**」をクリックし、ファイル・システム上の該当マクロの場所へナビゲートします。

注:

1. AID キーをホストに送信するアクションを実行する、終了画面を使用した Host On-Demand マクロが原因で、HATS ランタイムでの実行時にタイミングの問題が発生することがあります。マクロが実行を終了するときに、HATS ランタイムは終了画面をレンダリングします。しかし、AID キーが原因で、ホストは、レンダリングされていない別の (後続の) 画面に戻ります。このタイミングの問題を修正するには、以下のいずれかを行ってください。
   - 自動最新表示機能を使用して、後続の画面を表示します。 134 ページの『自動切断および最新表示 Web のみ』を参照してください。
   - Visual Macro Editor (VME) を使用して、マクロの終了画面を元の終了画面から後続の画面に変更します。VME の内蔵端末は Host On-Demand のように動作し、マクロの再生および後続画面の表示に使用できます。VME を使用して、後続画面をマクロに追加して、これが終了画面であることを示すようプロパティーを変更します。また、元の終了画面のプロパティーを変更して、終了画面の指示を除去します。詳しくは、*「HATS 拡張マクロ・ガイド」*の『Visual Macro Editor』の章を参照してください。
2. 固定反復ループ付きの Host Publisher マクロは、同一画面を継続的に認識し、その都度異なるアクションを実行します。HATS では、1 つの画面カスタマイズに

より、2 度目に同じ画面を認識し、一度目に認識したときとは異なるアクションを行うようにすることはできません。したがって、固定反復ループ付きの Host Publisher マクロを使用しようとすると、プロジェクトが無限ループに入ってしまうことがあります。固定反復ループ付きの Host Publisher マクロかどうかは、そのマクロのソース・コードを見れば分かります。この種のマクロには、ID 属性が `HPubFixedIterationLoop` である `customreco` タグと、ID 属性が `HPubIncrementLoop` である `custom` タグが含まれています。

マクロのエクスポート

HATS マクロだけでなく、IBM Host On-Demand マクロもエクスポートできます。これらのマクロをエクスポートするには、「**ファイル**」>「**エクスポート**」>「**HATS**」>「**HATS マクロ**」または「**ファイル**」>「**エクスポート**」>「**HATS**」>「**HOD マクロ**」を選択し、「**次へ**」をクリックします。

宛先を指定して「**エクスポートするマクロ・ファイル**」 を選択します。「**警告して既存のリソースを上書き (Overwrite existing resources with warning)**」チェック・ボックスをオンにすることによって、既存のリソースの上書きを選択できます。

実行したら「**完了**」をクリックします。

注:

DBCS サポートを使用する場合の考慮事項については、 549 ページの『マクロのエクスポート』を参照してください。

マクロのヒント

無限ループの回避

マクロの実行は、ホスト画面が画面カスタマイズの認識基準に一致した場合に実行するように指定できるアクションの 1 つです。マクロを記録する際は、最終画面が、そのマクロがアクションとして定義されている画面カスタマイズによって認識される画面とは別の画面であることを確認してください。認識された画面がマクロの最終画面である場合は、ループが作成されてしまいます。

一時画面の処理

マクロに対して非一時画面とともに一時画面を定義できます。ただし、実行時に、一時画面の 1 つが予期せずに認識されることがあります。

次の画面記述子のマクロを比較する場合、非一時画面記述子が最初に比較され、その後一時画面記述子が比較されます。非一時画面記述子の比較開始後に現行画面情報が更新されると、更新された画面情報が一時画面記述子の比較に使用されます。したがって、非一時画面の比較の開始時に画面がロックされ (OIA は入力禁止)、一時画面の比較が開始される前にその他の変更なしでアンロックされると、非一時画面が認識されなくなり、後続の一時画面を認識できる可能性がありますが、これは予期される動作ではありません。

この予期されない動作を回避するために、非一時画面と一時画面の記述子を比較するときに同じ画面情報を使用するかどうかを制御できます。これを行うには、拡張接続設定 **screenRecoUseOIASnapshot** を追加し、値を **true** に設定します。拡張接続設定の設定方法については、 161 ページのオプションの拡張接続設定を構成してくださいを参照してください。

第 12 章 グローバル変数との対話

グローバル変数には、1 つの HATS オブジェクトから他のオブジェクトに情報を渡すために使用できる値が入っています。例えば、ホスト画面上の複数の場所から情報を取り出し、計算を行い、その結果を現行画面または以後の画面に挿入することができます。1 つまたは複数のホスト画面からストリングの配列を作成し、変換に挿入することもできます。ユーザーが GUI 上のフィールドに入力したストリングを抽出して、別の場所で使用することもできます。

単一の HATS アプリケーションの有効範囲内でのみ可視であるグローバル変数は、ローカル・グローバル変数と呼ばれます。同じ .ear ファイルにあるすべての HATS Web アプリケーション、または同じリッチ・クライアント環境内で実行されているすべての HATS リッチ・クライアント・アプリケーションから可視であり、使用可能なグローバル変数は、共用グローバル変数と呼ばれます。グローバル変数を使用または作成する場合は、必ずこの変数がローカル変数または共用変数のいずれであるかを指定する必要があります。「**拡張**」設定セクションで共用を指定できます。すべてのグローバル変数のデフォルト設定は「**ローカル**」です。

注: 同じ名前を持つ 2 つのグローバル変数がある場合、1 つがローカルでもう 1 つが共用であれば共存が可能です。

HATS Web アプリケーションの場合は、単一 HTTP セッション中に同一 .ear ファイル 内のアプリケーション間でグローバル変数を共用できます。HATS リッチ・クライアント・アプリケーションの場合は、同一のリッチ・クライアント環境で実行されるアプリケーション間でグローバル変数を共用できます。リッチ・クライアント環境を閉じて再始動すると、以前に共用されていたグローバル変数値は保持されません。

グローバル変数の値は、数値、ストリング、またはストリングの索引付き配列です。グローバル変数にストリングの配列を含める場合は、配列全体を使用するか、特定の索引を使用するか、または特定の索引から始まるすべての値を使用するかを指定することができます。グローバル変数に対するすべての操作で、大文字と小文字が区別されます。グローバル変数には、「HATS」、「hats」、または「Hats」で始まる名前は使用しないでください。

グローバル変数の値を設定するには、次の方法があります。

- 画面イベントまたはアプリケーション・イベントに対して「**グローバル変数を設定**」または「**グローバル変数を抽出**」アクションを指定する。
- 画面変換でユーザーに値の入力を求めるプロンプトを表示する。
- マクロを実行しているときに、ユーザーに値の入力を求めるプロンプトを表示する。
- マクロ・ソース・コードで値を設定する。
- ビジネス・ロジック・プログラムの中で値を設定する。

グローバル変数に値が入っている場合、その値は以下の目的に使用できます。

© Copyright IBM Corp. 2003, 2015

- 「**グローバル変数を設定**」アクションで、他のグローバル変数の値を計算するため
- 「**データを挿入**」アクションを使用してホスト画面に値を書き込むため
- 「**グローバル変数を挿入**」メニュー項目を使用して、変換またはテンプレートに値を挿入するため
- マクロに値を渡すため
- ビジネス・ロジックの中で値を使用するため
- 画面認識で基準として使用するため
-

ホスト画面にグローバル変数を挿入する場合は、変換を適用する前にこのアクションをリストして、ホスト画面から作成された GUI にそのグローバル変数が表示されるようにする必要があります。ただし、変換が戻されてからホスト画面がホストに戻されるまでの間にグローバル変数を挿入した方が良い場合もあります。画面イベントのアクションを指定する方法の詳細については、181 ページの『アクション』を参照してください。グローバル変数を変換およびテンプレートに挿入する方法については、218 ページの『グローバル変数を挿入』を参照してください。

グローバル変数をプロンプト・マクロおよび抽出マクロで使用して、プロンプトに値を入力したり、ホスト画面から抽出した値を保管することができます。グローバル変数をマクロと連動して使用して、1 つの GUI で複数のホスト画面を結合できます。マクロでグローバル変数を使用する方法の詳細については、381 ページの『第 11 章 マクロとホスト端末』 および 415 ページの『第 14 章 画面の結合』を参照してください。

データを GUI からアプリケーションに受け渡すようにグローバル変数をオーバーライドすることができます。ローカル変数および共用グローバル変数は、オーバーライドできます。詳しくは、141 ページの『グローバル変数のオーバーライド』を参照してください。

これらは、画面認識にも使用できます。詳しくは、175 ページの『画面認識基準または開始画面』を参照してください。

ビジネス・ロジックでのグローバル変数の使用については、ご使用のアプリケーション環境に応じて、「*HATS Web アプリケーション・プログラマーズ・ガイド*」または「*HATS リッチ・クライアント・プラットフォーム・プログラマーズ・ガイド*」を参照してください。

グローバル変数を使用して複数の画面からのストリングまたは数値を累算する場合は、開始また接続アプリケーション・イベントに「**グローバル変数を設定**」アクションを追加することによって、そのグローバル変数を初期化することができます。

「**HATS Projects**」ビューで HATS プロジェクトの「**プロジェクト設定**」をダブルクリックします。「**イベント**」タブに進みます。

接続イベントのグローバル変数を設定するには、「**接続**」をダブルクリックして、イベント・エディターを開きます。次に、「**アクション**」タブでグローバル変数を追加できます。このようにして、実行されるアクションとグローバル変数をリンクできます。

同じ操作を**開始**イベントにも実行できます。

画面イベントまたはアプリケーション・イベントで「**グローバル変数を除去**」アクションを使用することにより、ローカルおよび共用グローバル変数を除去することができます。詳しくは、186 ページの『「グローバル変数を除去」アクション』を参照してください。

グローバル変数の名前変更

グローバル変数の各インスタンスは名前変更できます。特定の HATS プロジェクトにあるすべてのグローバル変数のリストを取得するには、以下のステップを実行します。

1. HATS メニュー・バーから「**ウィンドウ**」>「**ビューの表示**」>「**HATS グローバル変数**」を選択します。「**パレット**」ビューの直下に「**HATS グローバル変数**」ビューが配置されます。
2. 「**HATS Projects**」ビュー内のプロジェクトをクリックして、選択されたプロジェクトのマクロおよびイベント内にあるすべてのグローバル変数を「**HATS グローバル変数**」ビューに取り込みます。別のプロジェクトを選択すると、「**HATS グローバル変数**」ビューが更新されます。選択されたプロジェクトのマクロまたはイベントが追加、削除、または編集されると、内容が最新表示されます。
3. グローバル変数名を変更するには、「**HATS グローバル変数**」ビューでグローバル変数を右クリックし、変数名を編集します。マクロまたはイベントを開くには、各グローバル変数の下にリストされているマクロまたはイベントをダブルクリックします。マクロまたはイベントを作成または編集すると、「**HATS グローバル変数**」ビューが自動的に更新されます。

注: グローバル変数の名前変更は、マクロおよびイベント内で定義されたグローバル変数でのみサポートされます。したがって、「**HATS グローバル変数**」ビューにはこのようなグローバル変数のみが表示されます。グローバル変数を名前変更しても、ビジネス・ロジック、変換、または HATS プロジェクト内のほかの場所にあるグローバル変数は名前変更されません。

グローバル変数とマクロ変数の違い

HATS では、主に 2 つのタイプの変数があります。グローバル変数とマクロ変数です。ここでは 2 つの変数の違いを説明します。

グローバル変数

グローバル変数は、HATS Toolkit で作成され、HATS プロジェクトで使用される変数です。グローバル変数は、HATS の .hma ソース・ファイル内のマクロ・スクリプトの外部に格納されます。HATS ランタイムによって管理され、更新されます。グローバル変数には 2 つのタイプがあります。

ローカル
ローカル・グローバル変数は、HATS プロジェクト内で作成される変数であり、そのプロジェクトからのみ可視になります。

共用 共用グローバル変数は、.ear ファイル内のすべての Web アプリケーショ

ン、またはリッチ・クライアント環境のすべてのリッチ・クライアント・アプリケーションから可視であり、使用可能なグローバル変数です。

グローバル変数がローカル変数または共用変数のいずれであるかは、グローバル変数の作成時に GUI の「**共用**」チェック・ボックスが選択されたかどうか、または HATS の .hma ソース・ファイル内で **set**、**prompt**、または **extract** タグの共用属性の値が **yes** または **no** のいずれに指定されているかどうかによって決まります。

マクロ変数

マクロ変数は、 Visual Macro Editor または拡張マクロ・エディターで作成される変数です。マクロ変数はマクロ内部で使用されます。マクロの外部で使用されることはありません。マクロ変数は、マクロ・エンジンにより作成、保管、使用され、マクロ・スクリプトにリストされます。マクロ変数の詳細については、「*HATS 拡張マクロ・ガイド*」の『変数とインポートした Javaクラス』の章を参照してください。

HATS プロンプトを使用し、グローバル変数を抽出する HATS マクロ (.hma) ファイルのソースでは、プロンプトおよび抽出は、ソース・ファイル内のマクロ・スクリプトの前に組み込まれます。マクロ・スクリプトの構文は、<HAScript> タグと </HAScript> タグで囲みます。

第 13 章 統合オブジェクトの使用

この章では、ホスト・アプリケーションとの対話をカプセル化する Java Beans である、統合オブジェクトについて説明します。統合オブジェクトをサポートしているのは、HATS Web アプリケーションのみです。.

注: グローバル変数およびテンプレートは、HATS ポートレットで使用される統合オブジェクトではサポートされていません。

IBM WebSphere Host Publisher を使用したことがあれば、統合オブジェクトの性質の多くについては理解されているでしょうが、その場合でも、統合オブジェクトが HATS でどのように使用されるかを理解する必要があります。HATS で統合オブジェクトを作成したり使用するには、経験豊富な Host Publisher 開発者である必要はありません。

次の条件のいずれかに該当する場合には、統合オブジェクトを使用する必要がある場合があります。

- ご使用のアプリケーションが複数のホスト・アプリケーションに接続する。
- ホストとの対話を Enterprise JavaBeans (EJB) アプリケーションまたは Web サービスにカプセル化する必要がある。これらのプロセスの説明については、*「HATS Web アプリケーション・プログラマーズ・ガイド」*を参照してください。
- マクロの入力および出力に基づいて Web ページを構築する必要がある。

統合オブジェクトを作成すると、これを次のようなさまざまな用法で使用できます。

- ビジネス・ロジックから統合オブジェクトを実行する。
- 統合オブジェクトの入力および出力に基づいて、モデル 1、Struts、または JSF の Web ページを構築する。
- 統合オブジェクトをチェーニングして、統合オブジェクトを順番に実行できるようにする。
- 「URL に転送」アクションを実行して、変換指向の HATS プロジェクトから、1 つ以上のチェーニングされた統合オブジェクトを呼び出す JSP に制御を渡す。

統合オブジェクトを使用して実行できるその他の高度なタスクについては、*「HATS Web アプリケーション・プログラマーズ・ガイド」*を参照してください。次のようなタスクについての説明があります。

- 統合オブジェクトを使用して、ホスト・アプリケーションに含まれるビジネス・ロジックとトランザクションへの標準プログラミング・インターフェースを提供する、サービス指向アーキテクチャー (SOA) 資産を作成するために使用できる、従来型 (WSDL ベース) または RESTful の Web サービスを作成する。
- HATS EJB プロジェクトで統合オブジェクトを作成することにより、HATS Enterprise JavaBeans (EJB) アプリケーションを作成する。
- 統合オブジェクトの Java コードを変更する。
- 標準ポートレットでの統合オブジェクトの実行。

© Copyright IBM Corp. 2003, 2015

統合オブジェクトの作成

統合オブジェクトは、マクロから作成します。 381 ページの『第 11 章 マクロとホスト端末』のステップに従って、マクロを記録してください。

「HATS Projects」ビューでマクロを右クリックして、「統合オブジェクトの作成」を選択します。統合オブジェクトは、マクロと同じ名前になります。マクロを記録すると統合オブジェクトが自動的に生成されるように、HATS のプリファレンスを設定することもできます。「**ウィンドウ (Window)**」>「**プリファレンス**」をクリックし、左側のツリーで「**HATS**」を展開し、「**統合オブジェクト**」をクリックします。「**マクロ保存時に統合オブジェクトを自動的に生成**」というラベルの付いたボックスを選択して、「**適用**」をクリックします。

統合オブジェクトの作成に使用するマクロに命名するときは、必ず以下の要件に従ってください。

- マクロの名前には、英字、数字、または下線文字のみを使用できます。
- 名前の最初の文字を数字にすることはできません。

注:

1. 自動生成された統合オブジェクトは、チェーニングされていません。
2. DBCS サポートを使用する場合の統合オブジェクト作成については、 549 ページの『統合オブジェクトの作成』を参照してください。

チェーニングされたマクロから統合オブジェクトを作成することはできません。統合オブジェクト・チェーニングの場合、チェーン内の各統合オブジェクトの実行が完了してから次の統合オブジェクトに制御が渡されるので、マクロのチェーニングとはまったく異なります。マクロのチェーニングでは、HATS の「**マクロを実行**」アクションを使用して、現在のマクロ (「**マクロを実行**」アクションが発生したマクロ) を終了し、ターゲット・マクロの指定されたマクロ画面の処理を開始します。`<playmacro name="chainedMacroName"/>` タグを含む Host On-Demand マクロをインポートした場合、このマクロは HATS の「**マクロを実行**」アクションおよび「**マクロ・トランザクションを実行**」アクションのみに使用できます。マクロのチェーニングについて詳しくは、「*HATS 拡張マクロ・ガイド*」を参照してください。

あるマクロから統合オブジェクトが作成された後で、マクロ・エディターを使用してそのマクロを変更した場合、この変更を正確反映させるには、その統合オブジェクトを再生成する必要があります。「**マクロ保存時に統合オブジェクトを自動的に生成**」プリファレンスを選択した場合、統合オブジェクトは自動的に更新されます。

作成した統合オブジェクトのリストを表示するには、HATS Toolkit の「**HATS Projects**」表示タブをクリックし、プロジェクト名を展開します。次に、「**ソース**」>「**統合オブジェクト**」の順に移動し、フォルダーをダブルクリックして中身を表示します。

注: 「**ソース**」 の下にある「**統合オブジェクト**」フォルダーは、最初の統合オブジェクトが作成されるまでは表示されません。

統合オブジェクトのチェーニング

統合オブジェクト・チェーニングを使用すると、複数の統合オブジェクトを作成し、これらをご使用の Host Access Transformation Services アプリケーション内で単一のメジャー・タスクにグループ化できます。各統合オブジェクトが 1 つのサブタスクを実行し、統合オブジェクト・チェーンがメジャー・タスクを実行します。アプリケーションが実行されると、同じ接続を使用して、統合オブジェクトが順々に実行されます。

マクロを右マウス・ボタンでクリックして統合オブジェクトを作成するときに、「チェーニングされた統合オブジェクトの作成」を選択します。次に、その統合オブジェクトをチェーン内の最初のオブジェクトにするか、中間のオブジェクトにするか、最後のオブジェクトにするかを指定します。チェーン内には、複数の中間統合オブジェクトを置くことができます。この統合オブジェクトの開始状態および停止状態を示すラベルを入力します。チェーン内の最初の統合オブジェクトには開始状態ラベルがなく、チェーン内の最後の統合オブジェクトには停止状態ラベルがありません。状態ラベルは、チェーン内の統合オブジェクトが正しい順番で処理され、そのつど正しいデータが渡されることを保証する必要があります。

このセクションでは、統合オブジェクト・チェーニングについて詳しく説明し、HATS アプリケーション内で統合オブジェクト・チェーニングを使用する方法を示します。

どのようなときに統合オブジェクト・チェーニングを使用するかの判断

チェーン内にいくつの統合オブジェクトが必要かを判断するには、まず、実行する必要があるサブタスクをすべてリストします。例えば、*fileview* という名前のホスト・アプリケーションがあり、このホスト・アプリケーションを起動すると、ファイルのリストが表示されるとします。ユーザーは、ファイル名の横に「1」と入力し、**Enter** を押して、その中身を表示することにより、ファイルのリストから任意のファイルを選択できます。実行する作業は次の 2 つです。

1. ユーザーに表示するファイルのリストを取得する。
2. 選択されたファイルのファイル詳細を取得する。

2 つのマクロが必要です。なぜなら、2 番目のマクロを実行する前にユーザーの判断が必要なためです。2 番目のマクロは、ユーザー入力を待機する必要があります。ユーザー入力が必要な場所を定義するのが、複数のタスクを個々の統合オブジェクトに分離するための最初のステップです。

ユーザーの選択肢が複数ある場合、それぞれの選択によって引き起こされるホスト・アクションが異なるときは、それぞれの選択は別個の統合オブジェクトでなければなりません。その後、ユーザーの選択に基づいて、これらの統合オブジェクトを動的に組み合わせることができます。例えば、5 つの項目からなるメニューがあるとします。5 つの選択項目の中からユーザーが任意に選択できる場合、それぞれの選択は独立した統合オブジェクトでなければなりません。

アプリケーション内に別個のタスクがいくつあるかわかったら、アプリケーションに対するチェーニングの影響を判断できます。3 つ以上のタスクがある場合は、チ

ェーニングを使用することにより、アプリケーションの応答時間を改善したり、実行する必要があるマクロ記録の量を減らして統合オブジェクトの全体的な複雑さを減らすことができます。

統合オブジェクト・チェーニングの使用

チェーン内の統合オブジェクトは、接続を (ある画面で) ある状態にします。統合オブジェクトが実行を終了したら、その状態で (つまり、その画面で) 開始する別の統合オブジェクトが実行できます。

チェーニングを使用すると、複雑なアプリケーションを複数のタスクに分割できます。各タスクは 1 つの統合オブジェクトによって表現されます。統合オブジェクトの順序を正しく指定する必要があります。

例えば、あるチェーンに 3 つの統合オブジェクト (A、B、および C) がある場合、最初に A を使用し、次に B を使用し、最後に C を使用する必要があるとします。統合オブジェクト B よりも先に統合オブジェクト C が呼び出された場合、C がその接続を要求したときに接続が正しい状態で使用可能になっていないと、この統合オブジェクトは失敗します。

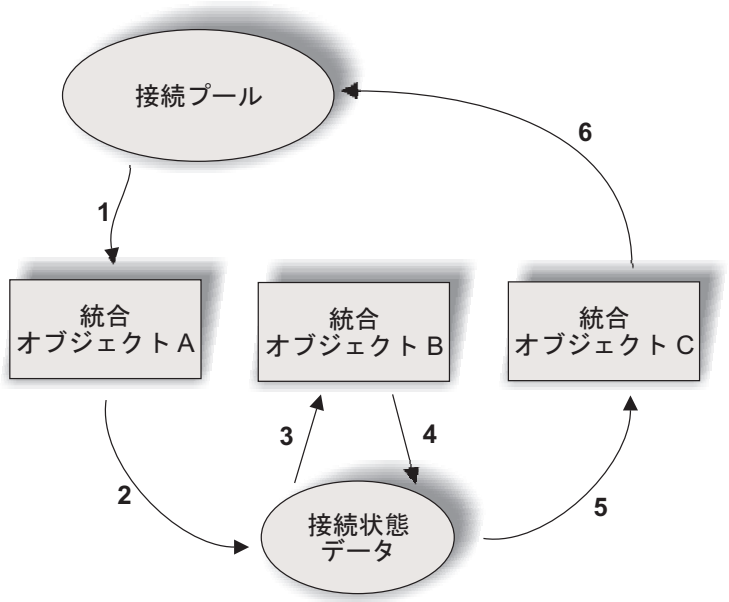


図 76. チェーニングを使用した場合の接続の存続時間

図 76 は、3 つの統合オブジェクトの実行全体を通しての接続の存続時間を示しています。統合オブジェクト A が最初で、統合オブジェクト B が中間で、統合オブジェクト C が最後として構成されています。接続状態データは、最後の統合オブジェクトの実行の接続および状態ラベルを表します。

1. 統合オブジェクト A は、実行を開始すると、マクロに指定されている接続を取得します。チェックイン画面が使用可能であり、かつ、接続が使用可能な場合、接続はすでにログオン済みでレディー状態です。チェックイン画面が使用可能で接続が使用不可である場合、接続が作成され、(指定されている場合は) 接続マク

ロが実行されます。チェックイン画面が使用不可の場合は、接続が作成され、(指定されている場合は) 接続マクロが実行されます。

2. 統合オブジェクト A は関連付けられたマクロを実行し、次の呼び出しに備えて、接続およびその現在の状態を保存します。(統合オブジェクト・チェーニング時に、この状態を停止状態ラベルとして定義しておきます)。
3. 統合オブジェクト B は、実行を開始すると、接続状態データから接続およびその状態を取得します。それはこれが中間の統合オブジェクトであるためです。統合オブジェクト B の開始状態ラベルは、統合オブジェクト A の停止状態ラベルとして定義しておく必要があります。これにより、これらの統合オブジェクト・チェーニングが可能になります。
4. 統合オブジェクト B の実行が完了すると、接続およびその状態が接続状態データに保存されます。
5. 統合オブジェクト C もまた、接続状態データから接続を取得します。統合オブジェクト C は、実行を開始すると、接続状態データから接続およびその状態を取得します。それはこれが最後の統合オブジェクトであるためです。統合オブジェクト C の開始状態ラベルは、統合オブジェクト B の停止状態ラベルとして定義しておく必要があります。これにより、これらの統合オブジェクト・チェーニングが可能になります。接続は常に戻されるので、チェーンの最後の統合オブジェクトには終了ラベルがありません。
6. チェックイン画面が使用可能である場合、接続はプールに戻ります。使用不可の場合、(指定されている場合は) 切断マクロが実行され、接続が終了します。

注: チェックイン画面は、接続設定の「**マクロ**」タブで設定できます。チェックイン画面設定を使用可能にするには、プールを使用可能にする必要があります。詳しくは、 158 ページの『接続エディター』を参照してください。

HATS で統合オブジェクト・チェーニングを使用してアプリケーションを構築するには、最初に統合オブジェクトを構築し、そのあとで、統合オブジェクトから Web ページを作成する必要があります。

以下のステップに従って、アプリケーションを作成してください。

1. マクロを記録します。
2. 各マクロから、チェーニングされた統合オブジェクトを作成します。
3. それぞれの統合オブジェクトの Web ページを作成

マクロの記録

HATS ホスト端末を使用して、HATS Toolkit にマクロを記録することができます。そのためには、HATS ホスト端末画面で、「**マクロの記録**」アイコンをクリックします。マクロの記録については、 385 ページの『マクロ関連アイコン』を参照してください。

作成するそれぞれの統合オブジェクトごとに、これらのステップを実行します。

それぞれのマクロからの統合オブジェクトの作成

すべてのマクロを作成後、HATS Toolkit の「**HATS プロジェクト表示**」タブに移動します。次に、プロジェクト名を展開し、「**マクロ**」フォルダーを開きます。

個々のマクロを右マウス・ボタンでクリックし、「**チェーニングされた統合オブジェクトの作成**」を選択します。オブジェクト・チェーン内での位置を、「**最初**」、「**中間**」、「**最後**」の中から選択できます。「**最初**」を選択した場合、統合オブジェクト・チェーニングの構成時の状態は停止状態になります。「**中間**」を選択した場合、状態は開始状態または停止状態になります。オブジェクト・チェーンの「**最後**」の位置では、開始状態を使用します。

チェーンを構成するすべての統合オブジェクトについて、この作業を行う必要があります。

それぞれの統合オブジェクトからの Web ページの作成

統合オブジェクトの作成が完了したら、Web ページを作成できます。統合オブジェクトから Web ページを作成する方法の詳細については、409 ページの『統合オブジェクトからの Web ページの構築』を参照してください。

統合オブジェクト・チェーニングを使用するアプリケーションのデバッグ

統合オブジェクト・チェーニングは、非常に複雑なホスト・アプリケーションをモデル化するための強力なツールですが、アプリケーションをアセンブルする際には注意が必要です。間違った順番で統合オブジェクトが呼び出される Java EE アプリケーションを作成してしまう恐れがあるため、このような事態が発生したときにサーバー上にどのようなエラーが表示されるかを認識している必要があります。

統合オブジェクトを間違った順番でチェーニングすると、統合オブジェクトを実行したときに問題が発生します。以下に、サーバー上に表示される可能性のあるエラー・メッセージおよびそのデバッグ方法を示します。

HPS5075 ファイル zzz.macro のマクロの再生時に STATE_PLAY_ERROR を受け取りました。

このエラーは、現在の画面と一致しないことが原因でマクロを再生できないときに発生します。これは、チェーン内の統合オブジェクトが正しい順番で呼び出された (使用する接続がそのチェーン内で見つかった) が、マクロ自体の実行が失敗した場合に発行される可能性があります。開始状態ラベルおよび停止状態ラベルがすべて同じ場合、あるいは、これらのラベルが 2 つ以上の統合オブジェクトに関して同じ場合、間違った論理順序で統合オブジェクトを呼び出すと、問題が発生します。

HPS5035 HttpSession の中にデータ・ソース・オブジェクト {0} がありません。考えられる原因としては、単一マシンから複数のブラウザーをチェーン・アプリケーションに対して使用している可能性があります。詳細については資料を参照してください。

このエラーは、最後の統合オブジェクト状態の開始状態ラベルを持つ統合オブジェクトが、そのオブジェクトの開始状態ラベルが直前の統合オブジェクトの停止状態ラベルと一致しなかったために、接続およびその状態の取得に失敗した場合に発生します。

この状況が発生した場合は、統合オブジェクトが間違った順番で呼び出された原因を判別してください。

- 別の統合オブジェクト (最初の統合オブジェクトまたは中間の統合オブジェクト) がすでに実行され、その接続が、最後の統合オブジェクト状態 (その停止状態ラベル) のラベルを持つ状態に置かれていますか。
- この統合オブジェクトを実行している JSP は正しい順番でリンクされていますか。
- 接続を最後の統合オブジェクト状態というラベルの付いた状態に置いた可能性のある、失敗した統合オブジェクトが JSP 上にありませんか。これにより、ページ上の別の統合オブジェクトの呼び出しが妨げられるとはかぎりません。一連のエラーである可能性があります。

統合オブジェクトからの Web ページの構築

任意の統合オブジェクトから JSP を作成できます。JSP を構築するには、モデル 1、JSF、および Struts という、3 つの方法があります。以下のセクションでは、これらの 3 つの方法について説明し、これらを使用してページを作成する方法を示します。

モデル 1 Web ページの作成

JSP を構築する HATS の従来の方法はモデル 1 と呼ばれます。

注: これは、HATS ポートレットでサポートされている唯一の方法です。

このモデルでは、単一の JSP に以下のものが含まれています。

- ユーザーに表示される情報
- 情報の表示方法を指定する書式設定タグ
- 次に呼び出されるページを制御する制御ロジック

「**HATS Projects**」ビューで、統合オブジェクトの名前を右クリックして、「**モデル 1 Web ページの作成**」を選択します。入出力ページ名と宛先を指定するか、あるいは統合オブジェクト名に基づいて作成されたデフォルト名を受け入れます。「**既存ファイルへの付加**」または「**既存ファイルの上書き**」を選択して、「**ページ生成オプション**」を指定することもできます。選択が終わったら、「**次へ**」をクリックします。

モデル 1 Web ページを作成する場合は、2 つのページを定義します。統合オブジェクトに必要なデータを収集する入力ページと、統合オブジェクトの実行後に結果を表示する出力ページです。

「**入力の定義**」ページで、各入力プロパティーの入力コントロールを定義できます。HTML コントロールがデフォルトとして使用されます。 コントロールを変更するには、入力プロパティーを選択し、「**編集**」をクリックします。オプションで、入力プロパティーを HTML コントロールまたは Dojo ウィジェットのいずれかを使用してレンダリングできます。同じモデル 1 Web ページで HTML コントロールと Dojo ウィジェットを混用できます。 コントロールに先行テキストを追加し、必要な場合は、選択したコントロールを構成できます。入力コントロールを使用する代わりに、入力プロパティーをグローバル変数の値に設定することを選択できます。オプションで、入力プロパティー値をグローバル変数として格納できます (統合オブジェクトが使用するグローバル変数は共用されます)。 入力ページを実行

すると、出力ページに制御が直接渡されるのではなく、HATS アプリケーションに制御が戻るため、これによってグローバル変数の設定やその他のアクションを必要に応じて行うことができます。

注:

1. マクロにプロンプトがない場合、入力ページは表示されません。
2. Dojo ウィジェットは HATS Web プロジェクトでのみ使用可能です。 HATS ポートレット・プロジェクトでは使用できません。

「**出力の定義**」ページで、表示する統合オブジェクトの出力プロパティーを選択します。コントロールを変更するには、出力プロパティーを選択し、「**編集**」をクリックします。オプションで、出力プロパティーを、HTML コントロール、Dojo ウィジェット、または通常テキストのいずれかを使用してレンダリングできます。同じモデル 1 Web ページで HTML コントロールと Dojo ウィジェットを混用できます。 コントロールに先行テキストを追加できます。 出力 JSP には、統合オブジェクトの初期化、グローバル変数からの統合オブジェクトのプロパティーの設定、統合オブジェクトの実行、および統合オブジェクト出力の表示を行うためのコードが含まれます。

ユーザーが作成するページは、「**HATS Projects**」ビューの「Web Content/Model 1 Pages」フォルダーに入ります。 Web ページの編集と Page Designer の起動を行うには、ページ名をダブルクリックします。ページをテストするには、入力ページ (入力がない場合は出力ページ) を右クリックし、「**サーバーでデバッグ**」または「**サーバーで実行**」のいずれかを選択します。

注: HATS ポートレット・プロジェクトで統合オブジェクトのモデル 1 Web ページを使用するときは、以下の考慮事項に注意してください。

- HATS 標準ポートレット・プロジェクトをコピーまたは名前変更すると、スタンドアロン統合オブジェクトまたはチェーン内の最初の統合オブジェクトから生成されたモデル 1 JSP は、誤った接続仕様を参照します。以下の記述を変更する必要があります。

  ```
  SignOn.setHPubStartPoolName("<new_project_name>/<default_connection>");
  ```

- Web プロジェクトからポートレット・プロジェクトを生成すると、以前に統合オブジェクトから生成されたモデル 1 JSP ページは変更されません。それらのページを再生成するか、手動で変更する必要があります。
- Dojo ウィジェットは HATS Web プロジェクトでのみ使用可能です。 HATS ポートレット・プロジェクトでは使用できません。

Struts Web ページの作成

Rational SDP の Struts ツール・フィーチャーをインストール済みの場合、HATS では、統合オブジェクトに基づく Web ページを作成するために Struts 2 を使用できるようにします。

注: この方法は HATS ポートレットではサポートされていません。

「**HATS Projects**」ビューで、統合オブジェクトの名前を右クリックして、「**Struts Web ページの作成**」を選択します。入力ページと出力ページの名前を指定するか、統合オブジェクト名に基づいて作成されるデフォルトの名前を受け入れます。「**完了 (Finish)**」をクリックする。

Struts Web ページを作成すると、次の新規ファイルが HATS により生成されます。

Struts アクション・クラス
このクラスはメイン・クラスです。このクラスは、統合オブジェクトのインスタンスを生成し、ActionForm クラスの値を使用して統合オブジェクトのプロパティーを設定し、統合オブジェクトのプール名を設定して、統合オブジェクトを実行します。次に、出力 JSP に転送します。あるいは、エラーがあった場合は、HATS ランタイムに戻り、HATS ランタイムにエラーの処理を委ねます。開発者は、このクラスを変更しないでください。このファイルは、プロジェクトの Source/Struts/Actions フォルダーにあります。

ActionForm クラス
このクラスには、入力 JSP の各入力の setter と getter が含まれています。入力 JSP は Struts サーブレットに実行依頼し、Struts サーブレットは ActionForm クラスのインスタンスを生成し、実行依頼された入力 JSP フォームに基づいてこのクラスのすべてのプロパティーを設定します。このクラスは、統合オブジェクトのすべてのプロパティーを設定する目的で、Action クラスによって使用されます。入力 JSP に入力プロパティーを追加したり、入力 JSP から入力プロパティーを除去した場合は、対応する setter および getter をこのファイルに追加 (または除去) する必要があります。このファイルは、プロジェクトの Source/Struts/ActionForms フォルダーにあります。

入力 JSP
入力 JSP には、Struts コントローラーに要求を送信する Struts HTML フォームが含まれています。Struts コントローラーは ActionForm を作成し、要求を適切な Action クラスに転送します。統合オブジェクトが入力データを必要としない場合、あるいは入力データが別のソース (例えば、グローバル変数) から取り込まれる場合、HTML フォームのみを含む空の入力 JSP が作成されます。入力ページは Web Content/Struts Pages フォルダー内に作成されます。ページをテストするには、ブラウザーでリンク `http://<hostname>:<port>/<contextRoot>/<StrutsActionClassName>.action` を実行します。

出力 JSP
出力 JSP は、統合オブジェクトの出力プロパティーを表示します。このページは、モデル 1 の出力 JSP と似ています。このファイルは、プロジェクトの Web Content/Struts Pages フォルダーにあります。

また、HATS は、Action クラスと入力/出力/エラー・ページを関連付けるために、struts.xml ファイルへのアクションの追加または更新を行います。

注: Struts がまだプロジェクトにインストールされていない場合は、Struts ページを作成するときに、HATS によって Struts がインストールされ、プロジェクトの source フォルダーの下に struts.xml ファイルが作成されます。

JSF Web ページの作成

JavaServer Faces (JSF) は、Java サーバーで実行する Web アプリケーション用のユーザー・インターフェース (UI) を開発するためのフレームワークです。再使用可能な UI コンポーネントをページでアセンブルすることにより、Web アプリケーショ

ンをすぐに構築できます。JSF Web ページを作成すると、Page Designer やパレットなどのツールを使用して変更や編集を行うことができます。

注: この方法は HATS ポートレットではサポートされていません。

「**HATS Projects**」ビューで、統合オブジェクトの名前を右クリックして、「**JSF Web ページの作成**」を選択します。入力ページと出力ページの名前、および宛先を指定するか、統合オブジェクト名に基づいて作成されたデフォルトの名前を受け入れます。「**既存ファイルの上書き**」チェック・ボックスを選択して、既存のファイルを上書きすることもできます。選択が終わったら、「**次へ**」をクリックします。

JSF Web ページを作成する場合は、2 つのページを定義します。統合オブジェクトに必要なデータを収集する入力ページと、統合オブジェクトの実行後に結果を表示する出力ページです。

入力ページの場合、各入力プロパティーの入力コントロールを定義できます。コントロールを変更するには、入力プロパティーを選択し、「**編集**」をクリックします。コントロールに先行テキストを追加し、必要な場合は、選択したコントロールを構成できます。「**次へ**」をクリックします。

出力ページの場合、表示する統合オブジェクトの出力プロパティーを選択します。コントロールを変更するには、出力プロパティーを選択し、「**編集**」をクリックします。コントロールに先行テキストを追加できます。「**完了**」をクリックします。

ユーザーが作成するページは、「**HATS Projects**」ビューの「Web Content/JSF Pages」フォルダーに入ります。Web ページの編集と Page Designer の起動を行うには、ページ名をダブルクリックします。ページをテストするには、入力ページ (入力がない場合は出力ページ) を右クリックし、「**サーバーでデバッグ**」または「**サーバーで実行**」のいずれかを選択します。

注:

BasicIOErrorPage.jsp および AdvancedIOErrorPage.jsp

BasicIOErrorPage は、統合オブジェクトを起動する JSP が使用するデフォルトのエラー・ページです。エラーが発生した場合、エラーの性質に関する基本的な情報がブラウザーに戻されます。この処理では、バックエンド接続は自動的に廃棄されます。

デフォルトの接続定義に基づいて接続が作成された場合、AdvancedIOErrorPage.jsp により、デフォルトの変換の HATS エントリー・サーブレットに接続が渡されます。この処理により、バックエンド接続との対話が可能になり、発生したエラーを特定できる可能性があります。デフォルトの変換の場合は、接続を切断することもできます。統合オブジェクトでエラーが発生した場合、基本的に AdvancedIOErrorPage.jsp により、ユーザーが処理を制御できる範囲が広がります。

これらのエラー・ページは、プロジェクトの「Web コンテンツ」フォルダーにある「**ナビゲーター**」ビューで参照できます。

JSP での統合オブジェクトの使用

以下に示す HATS ツールの機能は、JSP で統合オブジェクトを使用する場合に適用されます。以下に示す項目は、HATS Toolkit メニュー・バーの「**HATS ツール**」メニューから選択できます。

統合オブジェクト・プロパティーの挿入

統合オブジェクトの入出力を変換 JSP に追加できます。

入力

統合オブジェクトの入力を JSP に追加するときは、「**統合オブジェクト・プロパティーを挿入**」>「**入力**」を選択します。「統合オブジェクト・クラスの選択」ページで、使用する統合オブジェクトのクラス名、およびユーザーが入力フォームを送信するときに表示される宛先ページを入力します。「**次へ**」をクリックします。「入力の定義」ページで、コントロールを変更するには、入力プロパティーを選択し、「**編集**」をクリックします。 ここでは、先行テキストを定義したり、必要な場合は適切な入力コントロールを選択および構成したり、値をグローバル変数として保管したりできます。

注: Dojo ウィジェットは、モデル 1 Web ページに挿入される入力プロパティーのレンダリングにのみ使用できます。同じモデル 1 Web ページで HTML コントロールと Dojo ウィジェットを混用できます。

出力

統合オブジェクトの出力を JSP に追加するときは、「**統合オブジェクト・プロパティーを挿入**」>「**出力**」を選択します。「統合オブジェクト・クラスの選択」ページで、使用する統合オブジェクトのクラス名を入力し、チェック・ボックス「**統合オブジェクトのトランザクション・メソッドを挿入します**」をクリックして、統合オブジェクトを実行するトランザクション・メソッドを挿入します。「**次へ**」をクリックします。「出力の定義」ページで、コントロールを変更するには、出力プロパティーを選択し、「**編集**」をクリックします。 ここで、先行テキストの定義、および適切な出力コントロールの選択を行うことができます。

注: Dojo ウィジェットは、モデル 1 Web ページに挿入される出力プロパティーのレンダリングにのみ使用できます。同じモデル 1 Web ページで HTML コントロールと Dojo ウィジェットを混用できます。

HATS のグローバル変数には、ある HATS オブジェクトから別の HATS オブジェクトに情報を渡すために使用できる値が格納されます。統合オブジェクトの出力データを別の HATS オブジェクトに渡す場合は、統合オブジェクトを駆動するために使用する出力 JSP にコードを追加する必要があります。このコードは、統合オブジェクトが正常に実行された後で追加する必要があります。以下に例を示します。

```
<%
IOGV.setGlobalVariableString(session, "exampleGV",
exampleIO.getOutputData());
%>
```

この例では、共用グローバル変数 exampleGV に統合オブジェクト exampleIO の OutputData プロパティーの値が代入されます。

統合オブジェクト環境で使用するグローバル変数は、共用グローバル変数として定義する必要があります。この例のコードでは、共用グローバル変数が存在しない場合は、これが作成されます。

HATS アプリケーションへの転送の挿入

「**HATS アプリケーションへの転送の挿入**」は、JSP 制御情報をユーザーが起動した HATS アプリケーション (Web アプリケーションまたは標準ポートレットのいずれか) に戻すボタンを変換に追加するときに選択します。例えば、JSP によって統合オブジェクトが起動されて指定のタスクが実行されたら、HATS アプリケーションに制御を戻したい場合があります。このボタンの属性を変更するには、「**設計 (Design)**」ビューでボタンを強調表示し、右マウス・ボタンでクリックして「**属性**」を選択します。

第 14 章 画面の結合

一般的に、複数のホスト画面の出力データを収集して 1 つの出力ページに表示できるか、または複数のホスト画面に入力データを供給する 1 つの入力ページを作成できる必要があります。HATS では、このような操作を実行するための手法がいくつかあります。このような手法には、画面組み合わせ、画面カスタマイズ、変換、マクロ、グローバル変数、および統合オブジェクトなどの使用があります。使用する手法は、実行する操作とデータの場所によって異なります。

以降のセクションでは、4 種類の操作と、これらの操作を実行するために使用できる HATS の手法の要約を示します。操作と操作を実行する HATS 手法の組み合わせは、以下で説明する組み合わせに限りません。

連続出力データの結合

この例では、連続出力データとは、ホスト画面の 1 つの長方形領域にすべて表示され、後続の連続画面でも同じ場所に表示されるデータとして定義されています。この例としては、1 つの画面に収まらない出力データのリストがあります。この例では、すべてのデータを表示するために、ユーザーは端末エミュレーターを使用して、連続する画面を順方向にナビゲートする必要があります。ユーザーに対して表示される画面に関係なく、表示画面上でデータは同じ位置に表示されます。

このタイプのデータでは、HATS 画面組み合わせを使用して、複数画面からなる連続データのブロックを、ユーザーに対して表示される 1 つのスクロール可能な GUI にまとめることができます。画面組み合わせは HATS 画面イベントの一種です。画面イベントと画面組み合わせの作成および編集の詳細については、173 ページの『第 7 章 画面イベントの処理』を参照してください。

不連続出力データの結合

この例では、不連続出力データはホスト画面の複数の長方形領域に表示される出力データとして定義されており、同一のホスト・アプリケーションの他の (連続しているとは限らない) 画面では表示される場所が異なることがあります。この例としては、ホスト・アプリケーションの複数画面に分散しているデータがあります。

このようなデータを結合するには、HATS マクロを使用してホスト画面を自動的にナビゲートし、マクロの「**抽出アクションを追加**」機能を使用して、後で単一変換に表示できるようにデータをグローバル変数に取り込むか、または HATS マクロ・イベント・ハンドラー変換を使用してデータを表示します。HATS マクロの使用の詳細については、381 ページの『第 11 章 マクロとホスト端末』を参照してください。HATS グローバル変数の使用の詳細については、399 ページの『第 12 章 グローバル変数との対話』を参照してください。

© Copyright IBM Corp. 2003, 2015

複数アプリケーションからの出力の結合

この例では、出力データが、上記のように定義された連続出力データでも不連続出力データでも、複数のホスト・アプリケーションから出力されることを前提としています。これらのホスト・アプリケーションは、同一のホスト・システムまたは複数の異なるホスト・システムに存在していることがあります。

複数のホスト・アプリケーションからデータを収集するには、HATS 統合オブジェクトと、複数ホスト・アプリケーションへのバックグラウンド接続の HATS サポートを使用できます。統合オブジェクトとは、ホスト・アプリケーションとの対話をカプセル化する Java Bean です。これらは、HATS マクロから生成され、モデル 1 Web ページにより起動されます。

ホスト・アプリケーションごとにバックグラウンド接続、接続マクロ、切断マクロ、およびデータ・マクロを作成します。適切な入力を受け入れ、アプリケーションの画面をナビゲートし、ユーザーに対して表示するデータを抽出するデータ・マクロを作成します。各データ・マクロから統合オブジェクトを生成します。統合オブジェクトの 1 つからモデル 1 Web ページを生成します。これらのページでは、入力が統合オブジェクトに渡され、統合オブジェクトにより抽出された出力が表示されます。次に、他の統合オブジェクトを実行してその出力を同じ Web ページに追加するため、出力 Web ページを編集し、HATS ツールを使用して**統合オブジェクト・プロパティーを挿入**します。出力 Web ページの編集中に、複数の統合オブジェクト (複数のホスト・アプリケーション) からの出力を自由に編成し、ユーザーに対して表示するデータを結合します。HATS 統合オブジェクトの使用の詳細については、403 ページの『第 13 章 統合オブジェクトの使用』を参照してください。HATS 接続の定義については、157 ページの『第 6 章 接続の管理』を参照してください。

複数画面の入力の結合

同一アプリケーションの複数のホスト画面に入力データを提供する単一入力ページを作成するには、変換、グローバル変数、およびマクロを使用できます。例えば、変換を使用して入力データをグローバル変数に取り込むことができます。プロジェクト設定エディターの「**その他**」タブで使用可能な「**グローバル変数のオーバーライド**」機能を使用して、グローバル変数を初期化する変換に対してユーザーがデータを入力できるようにする方法があります。詳細については、141 ページの『グローバル変数のオーバーライド』を参照してください。次に、グローバル変数の値を入力として使用し、複数のホスト画面をナビゲートするマクロを作成し、ユーザーがマクロを実行するためにクリックできるボタンを変換上に配置します。

複数のホスト・アプリケーションに入力データを提供するには、『複数アプリケーションからの出力の結合』セクションで説明した方法に類似した、複数のバックグラウンド接続で実行する統合オブジェクトを使用する方法があります。統合オブジェクトを生成するマクロを作成するときに、「**プロンプト・アクションを追加**」機能を使用して、入力を求めるプロンプトを作成するか、または前述の「**グローバル変数のオーバーライド**」機能を使用できます。

第 15 章 印刷サポートの使用可能化

HATS プロジェクトを開発するときに、関連するホスト・アプリケーション用の印刷セッションを確立できます。 HATS アプリケーションを実稼働環境で実行中に、アプリケーションのユーザーは印刷用にフォーマットされたデータを印刷または表示することができます。印刷サポート設定は、デフォルトの HATS 接続だけに適用されます。

ホスト・アプリケーションと直接対話する場合、物理プリンターをアクティブ化してアプリケーションからデータを印刷します。HATS アプリケーションと対話する場合、物理プリンターはアクティブ化しません。代わりに、印刷ファイルが生成され、ユーザーはこのファイルを表示することができます。印刷ファイルは印刷することもできます。

注: 印刷ファイルが Adobe PDF ファイルで、PDF ビューアー (Adobe Acrobat Reader) がインストールされていない場合は、ファイルをディスクに保存するよう要求するプロンプトが出されます。

この章では、HATS プロジェクトで印刷サポートを使用可能にするプロセス、および展開済みアプリケーションの HATS Toolkit 環境とランタイム環境で印刷サポートを使用するプロセスについて説明します。

3270 ホストでのホスト印刷セッションの構成

HATS プロジェクトの印刷サポートを設定する前に、ユーザーまたはシステム管理者によってホスト印刷セッションの以下の構成が実行済みであることを確認してください。

- ディスプレイおよびプリンターの論理装置 (LU) を含む交換回線メジャー・ノードの VTAM® 構成。
- 印刷に使用するために関連するディスプレイおよびプリンター LU へのアクセスを提供する TN3270E ホスト・ポート に接続された HATS ホスト印刷アプリケーション。ディスプレイ/プリンター LU 関連は、TN3270E サーバー構成で指定および構成されます。

これらの手順の詳細については、3270 ホスト・ソフトウェアの資料を参照してください。

プロジェクト用印刷サポートの定義

プロジェクトで印刷サポートを利用するには、Telnet サーバーのホスト名とポート番号をシステム管理者から取得する必要があります。プロジェクト用の接続設定を定義するときに、これらの値を入力します。

3270E 接続の場合

3270E 接続の印刷サポートを定義するには、以下のステップに従ってください。

© Copyright IBM Corp. 2003, 2015

- ホスト上の印刷サポートが構成およびアクティブ化されていることをシステム管理者に確認します。

  注: HATS では、3270E 接続用に関連付けられた印刷のみをサポートします。つまり、エンド・ユーザーはディスプレイ LU またはディスプレイ LU プール名を指定し、印刷ジョブは割り当てられたディスプレイ LU に関連付けられたプリンター LU に送信される必要があります。
- プロジェクトで、デフォルト接続のホスト・タイプを 3270E と指定します。
- システム管理者からディスプレイ LU またはディスプレイ LU プール名を取得します。
- 3270E デフォルト接続用のエディターで、「**拡張**」タブをクリックし、「**LU またはプール名**」セクションで以下のオプションの 1 つを選択します。
  - ディスプレイ LU 名を使用する場合は、「**プロンプトを出す**」を選択するか、または「**指定値を使用**」を選択してディスプレイ LU 名を入力します。

    注: ディスプレイ LU 名を指定する場合は、複数のユーザーが同時にアプリケーションを使用するために、各ユーザーは接続時にディスプレイ LU 名をオーバーライドする必要があります。詳細については、130 ページの『接続パラメーターのオーバーライド』を参照してください。
  - ディスプレイ LU プール名を使用する場合は、「**指定値を使用**」を選択してディスプレイ LU プール名を入力します。
- 3270E デフォルト接続用のエディターで、「**印刷**」タブをクリックして「**印刷サポートを使用可能にする**」チェック・ボックスを選択します。
- 構成する印刷シナリオのタイプを選択します。事前定義された 3 つのスターター・シナリオから選択することも、拡張 Host On-Demand プリンター・セッション・プロパティーを使用して独自のシナリオを構成することもできます。3 つのスターター・シナリオとは、次のとおりです。
  - デフォルト印刷 (Adobe PDF 形式)
  - 基本テキスト・ファイル (プレーン・テキスト形式)
  - サーバーに接続されたデフォルトの Windows プリンターに送信 Web のみ

    または

    デフォルトの Windows プリンターに送信 RCP のみ
- 詳しくは、162 ページの『3270E 接続の印刷設定』を参照してください。
- VM システムで実行する場合は、印刷ジョブに使用するプリンターを確認します。

HATS アプリケーションのユーザーがファイルを印刷するコマンドを発行すると、ホスト・アプリケーションは印刷ジョブをプリンター LU に送信し、HATS ランタイムが印刷ジョブを接続用に構成されたフォーマットに変換します。ファイルのフォーマット後は、ユーザーが「**印刷ジョブを表示**」をクリックすると、キューに入れられた印刷ジョブのリストが表示されます。Web アプリケーションの場合、これはアプリケーション・キーパッド上のボタンです。リッチ・クライアント・アプリケーションの場合は、「アプリケーション」ビューのポップアップ・メニューの項目です。

注: 使用可能な印刷ジョブのリストはユーザーごとに個別に存在します。あるユーザーが印刷したジョブは、別のユーザーには表示されません。

該当するアイコンをクリックすることによって、キュー内のジョブのオープンまたは削除を行うことができます。例えば、PDF ファイルに対してオープンを選択すると、Adobe Acrobat などのビューアー・プログラムがインストールされている場合は、そのビューアー・プログラムで指定のファイルが開きます。この時点でファイルの印刷または保存を行うことができます。例えば、PDF ビューアーがインストールされていない場合は、ファイルを (.pdf ファイルとして) 保存するように要求されます。保存したファイルは、PDF ビューアー・プログラムがインストールされている別のシステムで開くことができます。

注:

1. HATS Web アプリケーションでは、印刷ファイルは WebSphere Application Server システムに保管されています。HATS アプリケーションを終了すると、キューにある印刷ファイルがすべて失われます。
2. HATS リッチ・クライアント・アプリケーションの場合、印刷ファイルはリッチ・クライアント・ワークステーション上に格納されています。ユーザーは、リッチ・クライアント環境を閉じる際にそれらのファイルを削除するかどうかを選択できます。その手順については、 100 ページの『印刷プリファレンス』を参照してください。
3. アプリケーションで、印刷ジョブを直接プリンターに送信するように構成されている場合は、印刷ファイルはどこにも保管されません。ユーザーはこのパネルでジョブを表示することも削除することもできませんが、ジョブが受信されたことは印刷が完了するまで確認できます。
4. 日本語ホスト・コード・ページ 1390 または 1399 を使用する 3270E 接続用に作成された PDF を表示するには、Adobe Reader 9 が必要です。JIS2004 文字を含まない、Adobe Reader 8 で表示可能な PDF ファイルを作成するには、3270E 接続に対する JIS2004 サポートを無効にする必要があります。これを行う方法については、 513 ページの『コード・ページ 1390 および 1399 用の JIS2004 サポートの無効化』を参照してください。

Host On-Demand PDT コンパイラーの使用

Host On-Demand PDT コンパイラーは、3270E ホスト印刷用の新規 Host On-Demand プリンター定義テーブル (PDT) ファイルを作成する必要があるユーザーの利便性を考慮して提供されるものです。PDT ファイルは通常は不要ですが、カスタマイズされた非 Windows の特定のプリンター・タイプ向けのバイナリー印刷出力ファイルを作成する必要がある場合に使用できます。PDT ファイルを作成するには、まずソースのプリンター定義ファイル (PDF) を作成または変更し、次にそれをバイナリー PDT ファイルにコンパイルします。PDT ファイルは Windows プリンターへの印刷には使用されません。

新規 PDT ファイルを作成するには、ソース PDF を <*RationalSDP_install_directory*>¥hats¥PDT¥pdfpdt¥usrpdf¥ ディレクトリーにコピーします。このディレクトリーには、ソース PDF のサンプルが提供されています。サンプル PDF を usrpdf ディレクトリーに unzip し、変更またはコンパイルしてください。<*RationalSDP_install_directory*>¥hats¥PDT¥ ディレクトリーにコマンド・プロンプトを開き、pdtcom.bat コマンドを実行します。「プリンター定義テーブルのコ

ンパイル (Compile a Printer Definition Table)」ウィンドウが開きます。このウィンドウのドロップダウン・ボックスからソース・ファイルを選択し、説明を入力して「**OK**」ボタンをクリックすると、そのファイルが PDT ファイルにコンパイルされます。コンパイルされた PDT ファイルは、<*RationalSDP_install_directory*>¥hats¥PDT¥pdfpdt¥ ディレクトリーに保管されます。コンパイル済み PDT ファイルのファイル拡張子は .hodpdt です。

コンパイル済みの PDT ファイルを、HATS Web アプリケーションの .ear プロジェクトのルートまたはご使用の HATS リッチ・クライアント・プロジェクトの HATS RCP ランタイム拡張プロジェクトのルートにあるフォルダー (例えば MyPDT)、あるいは、ご使用の HATS ポートレット・プロジェクトの WEB-INF フォルダー内にあるフォルダーにコピーします。

コンパイル済みの PDT ファイルは、デフォルト接続の接続エディターの「**印刷**」タブで、HATS に対して構成されます。「初期化」セクションで、「**基本テキスト・ファイル (プレーン・テキスト形式)**」を選択して「**初期化**」ボタンをクリックします。「名前/値」テーブルで、**PDTFile** プロパティーの値を **MyPDT/usribm3250.hodpdt** などに設定します。さらに、**printMimeType** プロパティーを **application/octet-stream** に、**usePDT** プロパティーを **true** に設定します。詳しくは、 162 ページの『印刷』を参照してください。ユーザーは、3270E 印刷出力を、特定タイプの非 Windows プリンターへのコピーに適したバイナリー印刷ファイルとして取得します。

Host On-Demand の PDF ファイルおよび PDT ファイルについて詳しくは、Host On-Demand Knowledge Center (http://publib.boulder.ibm.com/infocenter/hodhelp/v11r0/index.jsp?topic=/com.ibm.hod.doc/doc/troubleshoot/pdtcompile.html) を参照してください。

注:

1. プリンター定義ファイル (PDF) は、Adobe PDF ファイルとは何の関係もありません。
2. 他の HOD インストールに PDT ファイルが既に存在している場合は、それを再コンパイルせずにカスタム・プリンターに使用できます。その PDT ファイルを、前述のように Web プロジェクトまたはリッチ・クライアント・プロジェクトの適切な場所に単にコピーし、そのファイルを使用するようにデフォルト接続の印刷設定を設定してください。
3. 日本語 JIS2004 サポートの詳細については、512 ページの『3270E セッションの PDT 印刷および Print-to-File の JIS2004 サポート』を参照してください。

5250 接続の場合

5250 デフォルト接続用のエディターで 5250 接続用の印刷サポートを定義するには、「**印刷**」タブをクリックして「**印刷サポートを使用可能にする**」チェック・ボックスを選択します。System i Access for Web の「プリンター出力」ウィンドウの URL を指定します。デフォルトの URL は `http://`*`hostname`*`/webaccess/iWASpool` です。*hostname* は、5250 サーバーの名前です。5250 サーバーで印刷をサポートするには、印刷スプールがある 5250 ホストに System i Access for Web をインストールする必要があります。 5250 のすべての印刷機能は、ターゲットの 5250 ホストにある System i Access for Web のプリンター出力サーブレットによって提供されます。

その他の構成を行う必要はありません。HATS アプリケーションのユーザーがファイル印刷コマンドを発行すると、System i Access for Web によりホスト印刷ジョブが PDF 形式に変換され、ユーザー側にダウンロードしやすくなります。ユーザーは、アプリケーション・キーパッド上の「**印刷ジョブを表示**」をクリックすることで、System i Access for Web の「プリンター出力」ウィンドウを表示できます。このウィンドウで、次の印刷オプションを選択できます。

1. **PDF デバイス・タイプ (PDF device type)**
2. **用紙サイズ**
3. **宛先**

これらの印刷オプションについて詳しくは、System i Access for Web の資料を参照してください。

ユーザーへの資料の提供

ユーザーによる HATS アプリケーションの使用を容易にするには、HATS での印刷サポートの使用方法を説明した資料を提供することをお勧めします。その資料では、アプリケーションのグラフィカル・ユーザー・インターフェースまたはその他の簡単に表示できる形式 (テンプレート内でのリンク使用など) を使用して、次の項目を説明する必要があります。

- 以下に説明するような、「プリンター出力」ウィンドウでの機能の使用方法。
- PDF ビューアー (Adobe) がインストールされていない場合は、ファイルをディスクに保存するよう要求するプロンプトが出されます。

  注: ユーザーが Acrobat Reader の無料コピーをダウンロードするリンクをアプリケーションに追加することもできます。
- HATS アプリケーションを閉じると、自動的に「プリンター出力」ウィンドウが閉じます。
- 組み込む必要のあるアプリケーション固有の情報。

HATS 印刷サポートを使用するには、次の手順を実行する必要があります。

1. HATS アプリケーションを始動します。
2. ファイルを印刷します。
3. 「**印刷ジョブを表示**」をクリックします。Web アプリケーションの場合、これはアプリケーション・キーパッド上のボタンです。リッチ・クライアント・アプリケーションの場合は、「アプリケーション」ビューのポップアップ・メニューの項目です。印刷ジョブがある場合は、そのリストが「プリンター出力」ウィンドウに表示されます。

   「プリンター出力」ウィンドウで、ユーザーは、印刷ジョブのオプションをクリックして印刷ジョブを**オープン**または**削除** することができます。「オープン」をクリックすると、印刷ジョブは、Acrobat Reader が使用可能な場合 Acrobat Reader に PDF ファイルとして表示されます。Acrobat Reader が使用不可能な場合、ファイルをディスクに保存するよう要求するプロンプトが表示されます。

注:

a. 印刷ジョブのスプール中は、「プリンター出力」ウィンドウに印刷ジョブのファイル名が表示されることがありますが、**表示**および**削除**は PDF フォーマットへの変換が完了するまで使用できません。

b. 使用可能な印刷ジョブのリストはユーザーごとに個別に存在します。あるユーザーが印刷したジョブは、別のユーザーには表示されません。

第 16 章 キーボード・サポートの使用可能化

ユーザーは、ホスト・アプリケーションとの対話で、物理キーボード上の特殊キー (F1、Attn、Reset、Clear など) を頻繁に使用します。HATS アプリケーションのユーザーがキー・ストロークをホストに送信するには、以下のようにさまざまな方法があります。

- 物理キーボード上のキーを押す。この章で使用しているキーボード・サポート という用語は、このアクティビティーを指します。
- HATS のデフォルト・レンダリングによってレンダリングされるファンクション・キー用のボタンまたはリンクを使用する。ユーザーは、ボタンまたはリンクをクリックして、ホスト・キーをホストに送信します。
- 個々に挿入したホスト・キーパッド、ボタン、またはリンクを使用する。ユーザーは、ボタンまたはリンクをクリックして、ホスト・キーをホストに送信します。

この章では、HATS プロジェクトでキーボード・サポートを定義する方法について説明します。また、キーボード・サポートを文書化するためのヒントも示します。

HATS Web プロジェクトでキーボード・サポートを使用するには、サポートされる Web ブラウザーが必要です。サポートされる Web ブラウザーのリストおよび制限については、「Rational Host Access Transformation Services: System Requirements」(http://www.ibm.com/support/docview.wss?uid=swg27011794) および「Host Access Transformation Services 9.5 - Known issues and workarounds」(http://www.ibm.com/support/docview.wss?uid=swg27046998) を参照してください。

キーボード・サポートの定義

キーボード・サポートがデフォルトで有効になっていない場合は、「**プロジェクト設定**」の「**その他**」タブでキーボード・サポートを有効にして使用可能なオプションを選択できます。詳しくは、 129 ページの『キーボード・サポート』を参照してください。

また、ホスト・キーパッドとアプリケーション・アプリケーションを定義できます。詳細については、 122 ページの『ホスト・キーパッド』および 121 ページの『アプリケーション・キーパッド』を参照してください。

キーパッドの外観の変更

HATS Web プロジェクトの場合は、「**プロジェクト設定**」の「**レンダリング**」タブで「**ホスト・キーパッド**」と「**アプリケーション・キーパッド**」設定を使用して、表示するキーと、これらのキーをボタンまたはリンクのいずれとして表示するかを定義します。詳細については、 122 ページの『ホスト・キーパッド』および 121 ページの『アプリケーション・キーパッド』を参照してください。

特定の HATS Web プロジェクトのキーパッドのスタイルを変更するには、そのキーパッドに対応する Cascading Style Sheet (.css ファイル) を変更します。これは、

© Copyright IBM Corp. 2003, 2015

プロジェクトを構築する際に使用するスタイル・シートになります。Rational SDP ワークベンチで、HATS Toolkit の「**HATS Projects**」ビューに進み、プロジェクト名を展開します。 Web Content/Common/Stylesheets を展開します。詳しくは、374 ページの『スタイル・シートの使用』を参照してください。

スタイル・シートを変更するには、.css ファイルをダブルクリックしてエディターを開きます。テンプレートでは、スタイル・シートの変更をオーバーライドできます。Cascading Style Sheet の詳細については、http://www.w3.org/Style/CSS/ を参照してください。

HATS スタイルがまだ定義されていない場合は、スタイル・シートを使用して以下の任意の HATS スタイルを定義できます。

table.HostKeypad
ホスト・キーパッドの背景

table.ApplicationKeypad
アプリケーション・キーパッドの背景

input.HostPFKey
ホスト・キーパッドの PF ボタン

input.HostButton
ホスト・キーパッドのボタン

input.ApplicationButton
アプリケーション・キーパッドのボタン

a.HostKeyLink
ホスト・キーパッドのリンク

a.ApplicationKeyLink
アプリケーション・キーパッドのリンク

HATS リッチ・クライアント・プロジェクトの場合は、「**プロジェクト設定**」の「**レンダリング**」タブで「**ホスト・キーパッド**」設定を使用して、表示するキーと、これらのキーをボタンまたはリンクのいずれとして表示するかを定義します。詳細については、122 ページの『ホスト・キーパッド』を参照してください。表示するアプリケーション・レベルのボタンを定義するには、「**プロジェクト設定**」の「**レンダリング**」タブの「**アプリケーション・キーパッド**」を使用します。アプリケーション・レベルのボタンにテキストまたはイメージ、あるいはこの両方のいずれを表示するかを設定するには、「**レンダリング**」タブの「**ツールバー**」設定を使用します。詳細については、121 ページの『アプリケーション・キーパッド』および 121 ページの『ツールバー RCP のみ』を参照してください。

ユーザーへの資料の提供

HATS アプリケーションの使用を促進するために、ユーザー向けにキーボード設定に関する資料を作成することがあります。この資料は、アプリケーションのグラフィカル・ユーザー・インターフェースで作成するか、またはその他の簡単に表示できる形式 (テンプレートのリンクなど) で作成します。エンド・ユーザー用資料には、以下の事項に関する説明を含めてください。

- 「**キーボードのオン/オフ**」を使用してキーボード・サポートをオンまたはオフにする方法。
- HATS キーボード・マッピング (表 5 を参照)。
- Enter キーと HATS タブ付きフォルダーがどのように連動するのか。ユーザーが Tab キーで新しいタブに移動した後で Enter キーを押して新しいタブのコンテンツを表示するときは、キーボード・サポートはオフになっていなければなりません。キーボード・サポートがオフになっていないときに Enter キーを押すと、ホストに進むことになります。
- 必須レベルの Web ブラウザーのリストおよび HATS Web アプリケーションでのキーボード・サポート使用の制限については、「System Requirements for Host Access Transformation Services」(http://www.ibm.com/support/docview.wss?uid=swg27011794) および「Host Access Transformation Services 9.5 - Known issues and workarounds」(http://www.ibm.com/support/docview.wss?uid=swg27046998) を参照してください。

表 5. HATS でのキー・マッピング

ボタンまたはボタン・バー	**デフォルトの物理キー・マッピング**
F1 から F12	F1 から F12
F13 から F24	Shift + F1 から F12
ENTER	Enter
CLEAR	Esc
RESET	Ctrl + R
システム要求	Shift + Esc
ATTN	Pause/Break
前ページ	Page Up
次ページ	Page Down
PA1	Alt + Delete
PA2	Alt + End
PA3	Alt + Page Down
印刷	Ctrl + P Web のみ Ctrl + Shift + P RCP のみ
ヘルプ	Ctrl + H
キーボード使用可能/使用不可	Ctrl + K
デフォルト	Alt + Insert
最新表示	Alt + Page Up
HTML フォームのリセット	Ctrl + S
切断	Ctrl + D
印刷ジョブを表示	Ctrl + J
反転	Alt + Enter
フィールドの終了	Shift + Enter Web のみ テン・キーの Enter RCP のみ **注:** 詳しくは、 426 ページの『特殊な 5250 フィールド・キー・サポート』を参照してください。

表 5. *HATS* でのキー・マッピング *(続き)*

ボタンまたはボタン・バー	デフォルトの物理キー・マッピング
Field+	Ctrl + テン・キーのプラス (+) Web のみ テン・キーのプラス (+) RCP のみ
Field-	Ctrl + テン・キーのマイナス (-) Web のみ テン・キーのマイナス (-) RCP のみ **注:** 詳しくは、『特殊な 5250 フィールド・キー・サポート』を参照してください。

注: Ctrl+R は双方向セッションと非双方向セッションの両方のホスト RESET にマップされ、Alt+Enter は双方向セッションの反転にマップされています。

特殊な 5250 フィールド・キー・サポート

注: このようなモバイル・デバイスのサポートについては、54 ページの『モバイル装置に関する考慮事項および制約事項』を参照してください。

デフォルトのフィールド・キー押下の組み合わせ

HATS Web アプリケーションのデフォルトのフィールド・キー押下の組み合わせを以下にリストします。キーボードの再マップ方法について詳しくは、428 ページの『HATS アプリケーションでのキーボードの再マップ』を参照してください。

- フィールドの高速終了 (Quick Field Exit): Ctrl + Enter
- フィールドの終了: Shift + Enter
- フィールドの高速マイナス (Quick Field Minus): Shift + テン・キーのマイナス (-)
- フィールドのマイナス: Ctrl + テン・キーのマイナス (-)
- フィールドのプラス: Ctrl + テン・キーのプラス (+)

フィールドの高速終了 (Quick Field Exit)

HATS Web アプリケーションで「フィールドの高速終了 (Quick Field Exit)」キーの押下を使用することにより、5250 画面での数値データの入力を高速化することができます。このキー押下は、現行フィールドを現在のキャレット位置からフィールドの末尾まで消去し、カーソル位置を次のフィールドに進めます。このキー押下は任意のフィールドで使用できますが、ユーザーが「フィールドの終了」を使用する可能性がある場所で特に有用です。本来の「フィールドの終了」キーの押下は、データの画面をホストに送信し、HATS 変換を再描画させます。そのため、送信される前に多数の「フィールドの終了」援助機能を必要とする画面では、使用可能度の問題が発生する場合があります。「フィールドの高速終了 (Quick Field Exit)」キーの押下を「フィールドの終了エミュレーション (Field Exit Emulation)」フィーチャーと組み合わせて使用することにより、このようなアプリケーションの速度を大幅に高速化することができます。

リッチ・クライアント・アプリケーションは Web アプリケーションとは異なるアーキテクチャーを採用しているため、リッチ・クライアント・アプリケーションで

は「フィールドの高速終了 (Quick Field Exit)」キーの押下は必要ありません。リッチ・クライアント・アプリケーションでは、本来の「フィールドの終了」による往復遅延はありません。

フィールドの高速マイナス (Quick Field Minus)

HATS Web アプリケーションで「フィールドの高速マイナス (Quick Field Minus)」キーの押下を使用することにより、5250 画面での負の数値データの入力を高速化することができます。このキー押下は、現行フィールドを現在のキャレット位置からフィールドの末尾まで消去し、フィールド値の前に負符号文字 (-) を付加し、カーソル位置を次のフィールドに進めます。このキー押下は任意のフィールドで使用できますが、ユーザーが「フィールドのマイナス」を使用する可能性のある 5250 符号付き数値フィールドでのみ有効です。本来の「フィールドのマイナス」キーの押下は、データの画面をホストに送信し、変換を再描画させます。そのため、送信される前に多数の「フィールドのマイナス」援助機能を必要とする画面では、使用可能度の問題が発生する場合があります。「フィールドの高速マイナス (Quick Field Minus)」キーの押下を、「負の数値の入力の代替方法 (Alternate way to input negative numbers)」フィーチャーと組み合わせて使用することにより、このようなアプリケーションの速度を大幅に高速化することができます。5250 符号付き数値フィールド以外のフィールドで「フィールドの高速マイナス (Quick Field Minus)」キーの押下を使用すると、不適切なデータ入力が行われる可能性があります。

リッチ・クライアント・アプリケーションは Web アプリケーションとは異なるアーキテクチャーを採用しているため、リッチ・クライアント・アプリケーションでは「フィールドの高速マイナス (Quick Field Minus)」キーの押下は必要ありません。リッチ・クライアント・アプリケーションでは、本来の「フィールドのマイナス」による往復遅延はありません。

負の数値の入力の代替方法

5250 符号付き数値フィールドに負の数値を入力するには、「フィールドのマイナス」を使用してサーバーへの往復を実行する代わりに、負符号文字 (-) をフィールドの先頭位置に入力できます。これにより、多数の負の数値を 5250 の符号付き数値フィールドへより迅速に入力できるようになります。HATS サーバーにより、負符号文字 (-) を先頭に付けて入力した 5250 符号付き数値フィールドごとに「フィールドのマイナス」キーの押下がエミュレートされます。すべての不要な事前に入力されたデータは、別のフィールドにタブで移動する前にフィールドからクリアされる必要があります。負符号 (-) を先頭位置に入力しないでください。また、「フィールドのマイナス」、「フィールドのプラス」、または「フィールドの終了」も押さないでください。

フィールドの終了エミュレーション

このフィーチャーは、「フィールドの終了」援助機能キーを、必要に応じてユーザーに自動的に提供します。「フィールドの終了」援助機能キーは、特定のタイプの 5250 フィールドが変更されたときに必要となります。

例えば、ユーザーが「フィールドの高速終了 (Quick Field Exit)」キーの押下を使用してフィールドからフィールドへ移動し、Enter キーまたはその他の非フィールド援助機能キーを使用して画面を送信する場合、画面が送信されたときにカーソルを含むフィールドに対して、「フィールドの終了」が必要になる場合があります。

この場合、必要な「フィールドの終了」が提供されないと、ホストはエラー・メッセージ 「Enter キーはフィールドで使用できません (Enter key not allowed in field)」を表示する場合があります。 HATS サーバーでは、この HATS フィーチャーを使用して、Enter を送信する前に「フィールドの終了」をホストに送信し、このエラー・メッセージを回避します。

このフィーチャーを使用不可に設定する場合は、以下のステートメントを application.hap ファイルに追加してください。

```
<class name="com.ibm.hats.common.RuntimeSettings">
        <setting name="emulateFieldExit" value="false"/>
</class>
```

この名前の <class> セクションが既に存在する場合、<setting> 行だけをセクション内に追加してください。

HATS アプリケーションでのキーボードの再マップ

HATS キーを再マップする前に理解しておくべき概念

各 HATS アプリケーションは、Host On-Demand 製品のコードを組み込んで、クライアントに代わって Telnet サーバーへの接続を処理します。そのため、ユーザーがホスト・セッションでキーを押す操作を表し、ホスト・コンピューターに送信されるニーモニック・キーワードとして、このガイドでは Host On-Demand の用語を採用しています。

ニーモニック・キーワード

HATS でサポートされる 3270 ニーモニック・キーワード・テーブル

表 6. SendKeys のニーモニック・キーワード

機能	ニーモニック・キーワード
アテンション	[attn]
代替ビュー	[altview]
クリア	[clear]
Enter	[enter]
F1	[pf1]
F2	[pf2]
F3	[pf3]
F4	[pf4]
F5	[pf5]
F6	[pf6]
F7	[pf7]
F8	[pf8]
F9	[pf9]
F10	[pf10]
F11	[pf11]
F12	[pf12]

表 6. *SendKeys* のニーモニック・キーワード *(続き)*

機能	ニーモニック・キーワード
F13	[pf13]
F14	[pf14]
F15	[pf15]
F16	[pf16]
F17	[pf17]
F18	[pf18]
F19	[pf19]
F20	[pf20]
F21	[pf21]
F22	[pf22]
F23	[pf23]
F24	[pf24]
ホーム	[home]
PA1	[pa1]
PA2	[pa2]
PA3	[pa3]
リセット	[reset]
システム要求	[sysreq]

HATS でサポートされる 5250 ニーモニック・キーワード・テーブル

表 7. *SendKeys* のニーモニック・キーワード

機能	ニーモニック・キーワード
アテンション	[attn]
代替ビュー	[altview]
クリア	[clear]
Enter	[enter]
F1	[pf1]
F2	[pf2]
F3	[pf3]
F4	[pf4]
F5	[pf5]
F6	[pf6]
F7	[pf7]
F8	[pf8]
F9	[pf9]
F10	[pf10]
F11	[pf11]
F12	[pf12]
F13	[pf13]
F14	[pf14]

表 7. *SendKeys* のニーモニック・キーワード *(続き)*

機能	ニーモニック・キーワード
F15	[pf15]
F16	[pf16]
F17	[pf17]
F18	[pf18]
F19	[pf19]
F20	[pf20]
F21	[pf21]
F22	[pf22]
F23	[pf23]
F24	[pf24]
フィールドの終了	[fldext]
Field+	[field+]
Field-	[field-]
ヘルプ	[help]
PA1	[pa1]
PA2	[pa2]
PA3	[pa3]
システム要求	[sysreq]
Page Up	[pageup]
Page Down	[pagedn]
フィールドの高速終了 (Quick Field Exit)	[qfldext]
フィールドの高速マイナス (Quick Field Minus)	[qfld-]
リセット	[reset]
システム要求	[sysreq]
テスト要求	[test]

キー・コード値の判別

キーボード上のキーのキー・コード値を判別するには、次の HTML コードをファイルに保存し、Internet Explorer のブラウザー・ウィンドウで開きます。ブラウザーでキーを押すと、そのキーのキー・コード値が表示されます。

```
<html>
<head>
<script>
document.onkeydown = keydownhandler;
function keydownhandler()
{
 document.testform.displayField.value = "Keycode: " + window.event.keyCode;
}
</script>
</head>
<body>
 <center>
  <form name="testform">
   <input type="text" name="displayField" value="Hit any key">
```

```
  </form>
 </center>
</body>
</html>
```

HATS Web アプリケーションのキーの再マップ

すべての HATS Web アプリケーションには KBS.js というファイルがあります。このファイルは、ユーザーのキーボードにより生成されるキー・コード値と、ホスト・セッションへ送信されるキー・シグナルとのマップを指定します。HATS キーボードを再マップするには、KBS.js ファイルを編集する必要があります。KBS.js を見つけるには、「**ナビゲーター**」ビューに移動し、*project_name*/Web Content/Common ディレクトリーを検索します。KBS.js には、キーボードで押されたキーとホストへ送信されるシグナルの関係を決定する値が 3 つあります。3 つの値を次に示します。

1. 押されたキーのキー・コード値。これは、KBS.js ファイル内で整変数として示されます。例えば、以下のようになります。

   ```
   var CODE_F2 = 113;
   ```

   (F2 キーの場合の例)
2. キー・コード値とキーボードの Alt、Ctrl、Shift の組み合わせ。
3. キー・コードの値と Alt、Ctrl、Shift 状態の組み合わせについて送信される Host On-Demand ニーモニック・キーワード。例えば、以下のようになります。

   ```
   [CODE_F2, 0, 0, 0, '[pf2]']
   ```

   これにより、ユーザーのキーボードで F2 キーが押されたときに pf2 キー・シグナルがホストへ送信されます。

HATS Web アプリケーションでキー・マッピングを変更する場合は、新規キーまたは既存のキーがニーモニック・キーワードを送信できるように組み合わせを変更または追加します。

Ctrl、Alt、および Shift 修飾キーは再マップできない点に注意してください。HATS は JavaScript を使用し、JavaScript では、これらのキーはキープレスではなく修飾子とみなされます。

HATS Web アプリケーションでキーを再マップするために必要な処理は、KBS.js ファイルの上部でキー・コードと defaultKeyMappings 変数を変更することだけです。

注: KBS.js ファイルにリストされているキー、「プロジェクト設定」の「キーボード・サポート」の状態、画面にリストされているキー・ボタンの有無には、依存関係があります。この依存関係に関する規則を次に示します。

1. キーボード・サポートは、プロジェクト・レベルの設定を使用して設定し、HATS バージョン 5 以前の規則に従うかどうか、または表示されているボタンまたはリンクとは関係なくマップされたすべてのキーをサポートするかどうかを指定できます。これらの設定を変更するには、「**HATS Projects**」ビューを表示し、HATS プロジェクトの「**プロジェクト設定**」をダブルクリックし、「**その他**」タブを選択して「**キーボード・サポート**」をクリックします。詳しくは、129 ページの『キーボード・サポート』を参照してください。

2. デフォルトでは、HATS は、アプリケーション・キーパッドを表示しますが、ホスト・キーパッドは表示しません。HATS でホスト・キーパッドを表示するには、「**HATS Projects**」ビューを表示し、HATS プロジェクトの「**プロジェクト設定**」をダブルクリックし、「**レンダリング**」タブを選択して「**ホスト・キーパッド**」をクリックします。ここで、ホスト・キーパッドを表示するチェック・ボックスをオンにするかどうかと、どのキーを表示するかを選択できます。

例

キーボード上の Ctrl+Home キーの組み合わせをマップする方法の例を以下に示します。Home キーをホスト・セッションへ送信し、またこの機能用のボタンをご使用の変換に組み込むには、KBS.js を編集してから、 436 ページの『変換への追加のキーパッド・ボタンの追加』で説明するように、この変換を変更する 2 つのステップが必要です。

ホスト・キーを送信するキーの組み合わせを指定するには、以下に***太字イタリック***で示されている行を追加して KBS.js ファイルを変更します。

```
var CODE_BACKSPACE = 8;
 var CODE_TAB       = 9;
 var CODE_ENTER     = 13;
 var CODE_PAUSE     = 19;
 var CODE_ESC       = 27;
 var CODE_PAGEUP    = 33;
 var CODE_PAGEDOWN  = 34;
 var CODE_END       = 35;
 var CODE_HOME      = 36;
 var CODE_INSERT    = 45;
 var CODE_DELETE    = 46;
 var CODE_A         = 65;
 var CODE_B         = 66;
 var CODE_C         = 67;
 var CODE_D         = 68;
 var CODE_E         = 69;
 var CODE_F         = 70;
 var CODE_G         = 71;
 var CODE_H         = 72;
 var CODE_I         = 73;
 var CODE_J         = 74;
 var CODE_K         = 75;
 var CODE_L         = 76;
 var CODE_M         = 77;
 var CODE_N         = 78;
 var CODE_O         = 79;
 var CODE_P         = 80;
 var CODE_Q         = 81;
 var CODE_R         = 82;
 var CODE_S         = 83;
 var CODE_T         = 84;
 var CODE_U         = 85;
 var CODE_V         = 86;
 var CODE_W         = 87;
 var CODE_X         = 88;
 var CODE_Y         = 89;
 var CODE_Z         = 90;
 var CODE_HOSTHOME = 36;
 var CODE_F1        = 112;
 var CODE_F2        = 113;
 var CODE_F3        = 114;
 var CODE_F4        = 115;
 var CODE_F5        = 116;
 var CODE_F6        = 117;
 var CODE_F7        = 118;
```

```
var CODE_F8        = 119;
var CODE_F9        = 120;
var CODE_F10       = 121;
var CODE_F11       = 122;
var CODE_F12       = 123;
var HostKey        =1;
var ApplicationKey =2;
/*********************************************************
 * NOTICE: DO NOT MODIFY THE ABOVE VARIABLES!!
 *********************************************************/

var defaultKeyMappings = [
// KEYCODE,       ALT, CTRL, SHIFT, MNEMONIC
//============ command key mappings ================
  [CODE_ENTER,      0,    0,     0, '[enter]'     ],
  [CODE_PAUSE,      0,    0,     0, '[attn]'      ],
  [CODE_ESC,        0,    0,     0, '[clear]'     ],
  [CODE_ESC,        0,    0,     1, '[sysreq]'    ],
  [CODE_PAGEUP,     0,    0,     0, '[pageup]'    ],
  [CODE_PAGEUP,     1,    0,     0, 'refresh'     ],
  [CODE_PAGEDOWN,   0,    0,     0, '[pagedn]'    ],
  [CODE_HOSTHOME, 0,    1,     0, '[home]'    ],
  [CODE_PAGEDOWN,   1,    0,     0, '[pa3]'       ],
  [CODE_END,        1,    0,     0, '[pa2]'       ],
  [CODE_INSERT,     1,    0,     0, 'default'     ],
  [CODE_DELETE,     1,    0,     0, '[pa1]'       ],
  [CODE_D,          0,    1,     0, 'disconnect'  ],
  [CODE_H,          0,    1,     0, '[help]'      ],
  [CODE_P,          0,    1,     0, '[printhost]' ],
  [CODE_J,          0,    1,     0, 'printjobs'   ],
  [CODE_ENTER,      1,    0,     0, 'reverse'     ],
  [CODE_K,          0,    1,     0, 'toggle'      ],
  [CODE_S,          0,    1,     0, 'ResetButton' ],
//============ function key mappings ===============
  [CODE_F1,         0,    0,     0, '[pf1]'       ],
  [CODE_F1,         0,    0,     1, '[pf13]'      ],
  [CODE_F2,         0,    0,     0, '[pf2]'       ],
  [CODE_F2,         0,    0,     1, '[pf14]'      ],
  [CODE_F3,         0,    0,     0, '[pf3]'       ],
  [CODE_F3,         0,    0,     1, '[pf15]'      ],
  [CODE_F4,         0,    0,     0, '[pf4]'       ],
  [CODE_F4,         0,    0,     1, '[pf16]'      ],
  [CODE_F5,         0,    0,     0, '[pf5]'       ],
  [CODE_F5,         0,    0,     1, '[pf17]'      ],
  [CODE_F6,         0,    0,     0, '[pf6]'       ],
  [CODE_F6,         0,    0,     1, '[pf18]'      ],
  [CODE_F7,         0,    0,     0, '[pf7]'       ],
  [CODE_F7,         0,    0,     1, '[pf19]'      ],
  [CODE_F8,         0,    0,     0, '[pf8]'       ],
  [CODE_F8,         0,    0,     1, '[pf20]'      ],
  [CODE_F9,         0,    0,     0, '[pf9]'       ],
  [CODE_F9,         0,    0,     1, '[pf21]'      ],
  [CODE_F10,        0,    0,     0, '[pf10]'      ],
  [CODE_F10,        0,    0,     1, '[pf22]'      ],
  [CODE_F11,        0,    0,     0, '[pf11]'      ],
  [CODE_F11,        0,    0,     1, '[pf23]'      ],
  [CODE_F12,        0,    0,     0, '[pf12]'      ],
  [CODE_F12,        0,    0,     1, '[pf24]'      ]
];
```

注: 新規キー・コード変数 (この例では CODE_HOSTHOME) に付ける名前は、固有の名前で、defaultKeyMappings 変数リスト内の対応するエントリーと一致していれば、何でもかまいません。また、この例では Home キーのキー・コードに

```
var CODE_HOME = 36;
```

エントリーが既に存在しているため、実際には Home キー用の新規キー・コード変数を追加する必要はないことに注意してください。ただし、KBS.js ですでに定義されているキー・コード変数に関連しないキーボード・キーをマップする場合は、このようなエントリーが必要になります。

HATS リッチ・クライアント・アプリケーションのキーの再マップ

リッチ・クライアント環境では、デフォルトのキーボード・マッピングは HATS RCP ランタイム拡張プラグインの plugin.xml ファイルに定義されており、同じ環境で実行されるすべての HATS リッチ・クライアント・アプリケーションに適用されます。

常にユーザーに対して使用可能なコマンドを変更するには、Eclipse キーボード・コンテキストを使用します。ユーザーが HATS 変換ビューを使用している場合は、HATS キーボード・コンテキストが使用されます。このコンテキストは、HATS RCP ランタイム拡張プラグインの plugin.xml ファイルに登録されています。以下の例に、この記述を示します。

```
<extension
      point="org.eclipse.ui.contexts">
   <context
         name="%KEYBOARD_CONTEXT_NAME"
         description="%KEYBOARD_CONTEXT_NAME"
         id="com.ibm.hats.rcp.transformationContext"
         parentId="org.eclipse.ui.contexts.window">
   </context>
</extension>
```

デフォルトのキーボード・マッピングも plugin.xml ファイルで定義されています。以下の例は、Eclipse RCP または Lotus Expeditor で実行される HATS アプリケーション用のキー・マッピングがこのファイルでどのように記述されるかを示しています。

```
<extension
      point="org.eclipse.ui.commands">
   <category
         name="%COMMAND_CATEGORY_NAME"
         description"%COMMAND_CATEGORY_NAME"
         id="com.ibm.hats.rcp.transformationCategory"
   </category>
   <command
         name="[pf1]"
         category="com.ibm.hats.rcp.transformationCategory"
         id="com.ibm.hats.rcp.send_[pf1]">
   </command>
     ...
  </extension>
<extension
      point="org.eclipse.ui.bindings">
   <key
         commandId="com.ibm.hats.rcp.send_[pf1]"
         contextId="com.ibm.hats.rcp.transformationContext"
         configuration="org.eclipse.ui.defaultAcceleratorConfiguration"
         sequence="F1"
   </key>
     ...
 </extension>
```

Lotus Notes で実行するためのキー・バインディングは、その他のキー・バインディングとは別です。前の例に相当するキー・バインディングは、次のようになります。

```
<key
      commandId="com.ibm.hats.rcp.send_[pf1]"
      contextId="com.ibm.hats.rcp.transformationContext"
      schemeId="com.ibm.workplace.notes.hannoverConfiguration"
      sequence="F1"
</key>
```

例

Eclipse RCP または Lotus Expeditor で実行するリッチ・クライアント・アプリケーション用に、Ctrl+Home キーの組み合わせをキーボードにマップする方法の例を以下に示します。まず、以下に示すように [home] 用のコマンド・タグ、および Ctrl+Home 用のキー・タグを HATS RCP ランタイム拡張プラグイン内の plugin.xml ファイルに追加します。

```
<extension
      point="org.eclipse.ui.commands">
     ...
<command
      name="[home]"
      category="com.ibm.hats.rcp.transformationCategory"
      id="com.ibm.hats.rcp.send_[home]">
</command>
     ...
 </extension>

<extension
      point="org.eclipse.ui.bindings">
     ...
<key
      commandId="com.ibm.hats.rcp.send_[home]"
      contextId="com.ibm.hats.rcp.transformationContext"
      configuration="org.eclipse.ui.defaultAcceleratorConfiguration"
      sequence="Ctrl+Home"
</key>
     ...
 </extension>
```

次に、変換ビューにより新規キー・マッピングが plugin.xml ファイルに登録されたことが認識されるように、変換ビューのコードを更新します。この更新を行うには、ご使用のプロジェクト内の MainView.java ファイルを編集します。このファイルは、「HATS Projects」ビューのプロジェクトの下にある ¥Source¥*<project name>*¥views フォルダー内にあります。MainView.java ファイルをダブルクリックして、エディターで開きます。MainView メソッドの次に、以下に示す createKeyboardActions メソッドを追加します。

```
protected void createKeyboardActions() {
    super.createKeyboardActions();
    addKeyboardAction("[home]", "com.ibm.hats.rcp.send_[home]");
}
```

HATS Web アプリケーションと同様に、HATS リッチ・クライアント・アプリケーションでも Ctrl、Alt、および Shift 修飾キーは再マップできない点に注意してください。

ユーザーがデフォルト Eclipse キーボード設定を構成するための手段を提供できます。これは通常、メニュー・バーから「**ウィンドウ**」>「**設定**」>「**ワークベンチ**」>

「**キー**」を使用して構成します。ユーザーに対してこのような手段を提供すると、ユーザーは HATS RCP ランタイム拡張プラグインで定義されているキー・マッピングをオーバーライドできます。

変換への追加のキーパッド・ボタンの追加

デフォルトのホスト・キーパッドに新規ボタンまたはリンクを追加できます。これは、「**プロジェクト設定**」の「**レンダリング**」タブの「**ホスト・キーパッド**」設定で制御されます。新しいボタンまたはリンクは、デフォルトのホスト・キーパッドが使用されるたびに表示されます。単一の変換に新しいボタンまたはリンクを追加するには、変換の編集時に「**ホスト・キーパッドを挿入**」>「**個別キー**」機能を使用します。詳細については、219 ページの『個々のキー』を参照してください。独自のボタンまたはリンクを追加するための GUI には、表題とニーモニックが必要です。この例では、表題が **Home** で、ニーモニックが **[home]** です。この新しいボタンに対するキーボード・サポートが必要な場合、KBS.js defaultKeyMappings テーブル (Web アプリケーション) または HATS RCP ランタイム拡張プラグインの plugin.xml ファイル (リッチ・クライアント・アプリケーション) にそのボタンを追加する必要があります。

第 17 章 ホスト・シミュレーションの使用

HATS Toolkit には、ホスト・シミュレーション機能があります。この機能を使用して、ホスト・シミュレーション・トレース・ファイルを記録できます。このホスト・シミュレーション・トレース・ファイル・ファイルは、保存してライブ・ホスト接続の代わりに使用できます。ライブ・ホスト接続を使用する代わりにホスト・シミュレーション・ファイルを使用する操作を以下に示します。

- ホスト端末機能を使用した画面キャプチャー、画面イベント、および画面変換の作成。
- ホスト端末機能を使用したマクロの作成とテスト。
- Rational SDP ローカル・テスト環境を使用して、HATS アプリケーションをテストします。
- デモンストレーションに使用する HATS アプリケーションのランタイム環境への展開。

さらに、失敗したシナリオを記述する目的で、ランタイム環境で他のトレースおよびログとともにホスト・シミュレーション・ファイルを記録できるようにすることで、ホスト・シミュレーション機能をトラブルシューティングに使用できます。

注:

1. ホスト・シミュレーションは、SSL 対応接続ではサポートされていません。
2. スタンドアロンのホスト・シミュレーター・ツールで記録された ENPTUI 対応デバイスのホスト・シミュレーション・トレースは、ENPTUI が HATS プロジェクトで無効になっている場合は正しく再生されないことがあります。

「ホスト・シミュレーション」ウィザード

「ホスト・シミュレーション」ウィザードを開始するには、以下のステップのいずれかを実行します。

- メニュー・バーから、「**ファイル**」>「**新規**」>「**HATS ホスト・シミュレーション**」を選択します。
- メニュー・バーから、「**HATS**」>「**新規**」>「**ホスト・シミュレーション**」を選択します。
- ツールバーから、「**HATS ホスト・シミュレーション・ファイルを作成**」項目をクリックします。
- HATS プロジェクトを右クリックし、「**新規 HATS**」>「**ホスト・シミュレーション**」を選択します。
- HATS EJB プロジェクトを右クリックし、「**新規ホスト・シミュレーション (New Host Simulation)**」を選択します。

ウィザードの 1 番目のページである「**ホスト・シミュレーション・トレースを作成**」ページには、次のフィールドがあります。

© Copyright IBM Corp. 2003, 2015

プロジェクト
このフィールドには、ホスト・シミュレーション・トレース・ファイルのターゲット・プロジェクトを指定します。

名前 このフィールドには、ホスト・シミュレーション・トレース・ファイルの名前を指定します。デフォルト名は ProjectNameTrace_num です。ProjectName はターゲット・プロジェクトの名前、num は基本ファイル名がすでに使用されている場合に付加される番号です。

場所 このフィールドには、ホスト・シミュレーション・トレース・ファイルの宛先の場所が表示されます。これは変更できません。

説明 オプションの説明を追加するときに、このフィールドを使用します。

ウィザードの 2 番目のページは「**接続の選択 (Choose Connection)**」ページです。このページでは、プロジェクトでトレース・ファイルの記録に使用される接続を選択します。プロジェクトのデフォルト接続が記録のデフォルト接続になります。

ウィザードで「**完了**」をクリックすると、ホスト・シミュレーション・レコーダーが「**HATS ホスト・シミュレーション**」プリファレンスの「**TCP/IP ポートの範囲**」設定から使用する未使用のポートを検出し、レコーダー自体を実際のホストと HATS ホスト端末の間のプロキシーとしてセットアップします。詳細については、149 ページの『HATS プリファレンスの使用』を参照してください。次にホスト端末が表示されます。

ホスト端末を使用して、このトレース・ファイルのシナリオを記録するためにホスト・システムと対話します。記録を停止するには「**ホスト・シミュレーションの記録を停止**」ツールバー・アイコンを使用し、記録を再開するには「**ホスト・シミュレーションの記録を再開**」ツールバー・アイコンを使用します。

記録を停止すると、トレース・ファイルを保存するオプションと、ホスト端末ウィンドウを閉じるかどうかを選択するオプションが示されます。保存されたトレース・ファイルは、「HATS Projects」ビューでターゲット・プロジェクトの下の「**ホスト・シミュレーション**」フォルダーに表示されます。

ホスト・シミュレーション・エディター

ホスト・シミュレーション・エディターでホスト・シミュレーション・トレース・ファイルを開くには、プロジェクトの「**ホスト・シミュレーション**」フォルダーの中にあるこのファイルをダブルクリックします。このエディターでは、トレース情報を参照し、再生設定を変更することができます。ホスト・シミュレーション・エディターには、「**概要**」タブと「**ソース**」タブの 2 つがあります。

「概要」タブ

「**概要**」タブには、「**一般情報**」セクションと「**再生設定**」セクションの 2 つがあります。

「一般情報」セクション

「一般情報」セクションには、トレース・ファイルの名前と説明、最終変更日時、および関連付けられている接続の名前が表示されます。

「再生設定」セクション

このセクションでは、再生設定を変更し、ホスト端末でトレースを再生するボタンをクリックできます。変更できる再生設定を以下に示します。

TCP/IP ポートの範囲

トレース再生中に、HATS ホスト・シミュレーション機能はホスト端末または HATS アプリケーションとのホスト通信をシミュレートするため、特定の範囲のポートを listen します。複数のセッションで同時にトレース再生を実行できるように、複数ポートが使用されます。HATS ホスト・シミュレーション設定を使用して、デフォルト値が事前に設定されています。詳細については、149 ページの『HATS プリファレンスの使用』を参照してください。

時間遅延設定

「**遅延**」、「**最小 (ms)**」、および「**最大 (ms)**」の値を使用して、ホスト・シミュレーションが再生中にホスト端末または HATS アプリケーションからの要求応答を待機する遅延時間 (ミリ秒単位) を設定します。HATS ホスト・シミュレーション設定を使用して、デフォルト値が事前に設定されています。デフォルト値はここで変更できます。 詳細については、149 ページの『HATS プリファレンスの使用』を参照してください。

「ホスト端末上で再生」ボタン

ホスト端末ウィンドウを開いてホスト・シミュレーション・トレース・ファイルを再生するには、このボタンをクリックします。

「ソース」タブ

「**ソース**」タブでは、XML エディターにホスト・シミュレーション・トレース・ファイルのソースが表示されます。

実行時環境での記録

Web 環境では、管理者は HATS 管理コンソールを使用して実行時のホスト・シミュレーション記録を使用可能にできます。その手順については、452 ページの『トレース・オプションの設定』を参照してください。

リッチ・クライアント環境では、ユーザーがワークベンチ・プリファレンスを使用して、実行時のホスト・シミュレーション記録を使用可能にできます。その手順については、100 ページの『トラブルシューティング・プリファレンス』を参照してください。

再生オプション

各種ホスト・シミュレーション再生オプションの要約を次の表に示します。

表 8. ホスト・シミュレーション再生オプション

環境	起動アクション	機能
ホスト端末	「HATS Projects」ビューでホスト・シミュレーション・ファイルを右クリックし、「**ホスト端末上で再生**」を選択します。	再生表示します。

表 8. ホスト・シミュレーション再生オプション (続き)

環境	起動アクション	機能
ホスト端末	「HATS Projects」ビューで、ホスト・シミュレーション・ファイルをダブルクリックします。ホスト・シミュレーション・ファイル・エディターで「**ホスト端末上で再生**」ボタンをクリックします。	再生表示します。
ホスト端末	「HATS Projects」ビューで、接続をダブルクリックします。接続エディターの「**基本**」タブで、「**ライブ接続の代わりに、ホスト・シミュレーション・ファイルを使用**」を選択し、ドロップダウンからトレースを選択します。 「HATS Projects」ビューで接続を右クリックし、「**ホスト端末をオープン**」を選択します。	画面キャプチャー、画面イベント、および画面変換を作成します。 マクロを作成してテストします。
ローカル・テスト・ランタイム (Web、EJB、およびリッチ・クライアント・アプリケーション)	「HATS Projects」ビューで、接続をダブルクリックします。接続エディターの「**基本**」タブで、「**ライブ接続の代わりに、ホスト・シミュレーション・ファイルを使用**」を選択し、ドロップダウンからトレースを選択します。 アプリケーションをローカル・テスト環境で実行します。	HATS アプリケーションをローカル・テスト環境でテストします。
展開済みランタイム (Web、EJB、およびリッチ・クライアント・アプリケーション)	「HATS Projects」ビューで、接続をダブルクリックします。接続エディターの「**基本**」タブで、「**ライブ接続の代わりに、ホスト・シミュレーション・ファイルを使用**」を選択し、ドロップダウンからトレースを選択します。 アプリケーションをエクスポートおよび展開します。	HATS アプリケーションをデモンストレーションとして展開します。

トレース・ファイルのインポートおよびエクスポート

サポート対象外のスタンドアロン・ツールであるホスト・シミュレーターが、ダウンロード可能でした。ホスト・シミュレーターにより、HATS V9.5 に組み込まれているホスト・シミュレーション機能により今は使用可能となっている機能の一部をユーザーが実行できました。スタンドアロンのホスト・シミュレーター・ツールにより作成されるトレース・ファイルと、HATS ホスト・シミュレーション機能により作成されるトレース・ファイルの両方をインポートできます。スタンドアロンのホスト・シミュレーター・ツールにより作成されたトレース・ファイルをインポートするには、メニュー・バーから「**ファイル**」>「**インポート**」
>「**HATS**」>「**HATS へのホスト・シミュレーター・トレース**」を選択します。
HATS ホスト・シミュレーション機能により作成されたトレース・ファイルをインポートするには、メニュー・バーから「**ファイル**」>「**インポート**」

>「HATS」>「HATS ホスト・シミュレーション」を選択します。ウィザードで**「追加」**ボタンをクリックして、ファイル・システム内を参照し、インポートするトレース・ファイルを選択します。次に宛先プロジェクト内で「ホスト・シミュレーション」フォルダーを選択します。スタンドアロンのホスト・シミュレーター・ツールにより作成されたトレース・ファイルをインポートする場合は、プロジェクト内でこのトレース・ファイルに関連付ける接続を選択します。**「完了」**をクリックします。

HATS ホスト・シミュレーションのトレース・ファイルをエクスポートするには、メニュー・バーから**「ファイル」>「エクスポート」>「HATS」>「HATS ホスト・シミュレーション」**を選択します。次に、エクスポートするトレース・ファイルを選択し、参照して宛先を選択し、**「完了」**をクリックします。

第 18 章 HATS 管理コンソールの使用

HATS 管理インターフェースを使用して、接続を管理し、HATS Web アプリケーションの問題を判別できます。この章では、これらの管理機能の使用方法について説明します。HATS リッチ・クライアント・アプリケーションに関する情報については、93 ページの『HATS リッチ・クライアント・アプリケーションの管理』を参照してください。

WebSphere Application Server に展開されている Enterprise Archive (.ear) ファイルに含まれる一連の HATS アプリケーションに HATS 管理コンソールのサポート・ファイルを含めることができます。展開済み .ear ファイルはランタイム環境を提供します。この環境では、HATS 管理コンソールを使用することにより HATS アプリケーションを管理できます。複数のプロジェクトに HATS 管理コンソール・ファイルを含めると、異なるアプリケーション名を使用して複数の管理コンソールを開始できます。ただし、各管理コンソールは同一のビューと機能を提供します。

注: このオプションは、HATS ポートレット・プロジェクトではサポートされていません。

各 .ear ファイルに HATS 管理コンソール・サポート・ファイルを含める代わりに、HATS 管理コンソール・ファイルのみを含む .ear ファイルを作成することができます。この .ear ファイルを展開する場合は、HATS 管理コンソールを開始して管理の有効範囲を変更することで、1 つの管理コンソールのインスタンスで複数の .ear ファイルの HATS アプリケーションをリモート側で管理できます。展開済み HATS 管理コンソール・アプリケーションにより、すべての HATS アプリケーションをまとめて 1 箇所から管理できます。新規 HATS プロジェクトを作成するときに、プロジェクトに HATS 管理コンソール・ファイルを含めるかどうかを指定します。HATS 管理コンソール・プロジェクトを作成する場合、作成したほかのプロジェクトに管理サポート・ファイルを追加する必要はありません。

HATS Toolkit で HATS 管理コンソール・プロジェクトを作成するには、「**ファイル**」>「**新規**」>「**プロジェクト**」>「**HATS**」>「**HATS 管理プロジェクト**」を選択します。このプロジェクトを HATS 管理コンソール・プロジェクトとして一意に識別する Enterprise Application プロジェクト名を指定します。HATS 管理コンソール・プロジェクトは、HATS Toolkit の「**ナビゲーター**」タブにのみ表示されます。作成した .ear ファイルを WebSphere Application Server に展開して開始します(447 ページの『HATS 管理コンソールの開始』を参照)。HATS 管理コンソール・プロジェクトの管理の有効範囲を変更する場合には、448 ページの『管理の有効範囲の選択』を参照してください。

その他の管理トピックについては、454 ページの『端末の表示機能』で説明します。

© Copyright IBM Corp. 2003, 2015

HATS 管理コンソールおよび WebSphere セキュリティー

HATS 管理コンソールでは、Java Management Extensions (JMX) Bean を使用してリモート管理が実行されます。WebSphere Application Server のグローバル・セキュリティーが有効になっている場合は、これらの JMX 呼び出しが WebSphere セキュリティーにより認証されるため、WebSphere で有効な許可レベルが設定されている必要があります。したがって、ユーザー ID が、管理者権限またはオペレーター権限が付与されている WebSphere Application Server 管理コンソール・ユーザーとして定義されていないと、HATS 管理コンソールから適切にリモート管理操作を実行できません。

HATS 管理コンソールでは、問題判別設定を表示および変更できます。また、接続状況を確認し、ホスト接続を切断できます。HATS 管理機能を使用して HATS アプリケーションを展開する場合は、セキュリティーのために、HATS の 3 つの役割をそれぞれ特定のシステム・ユーザー ID にマップできます。定義されている 3 つの HATS 役割 (HATSAdministrator、HATSOperator、および HATSMonitor) では、それぞれに HATS 管理コンソールで実行できる機能が異なります。詳しくは、446 ページの『HATS 管理コンソールの役割』を参照してください。

同様に、WebSphere Application Server 管理コンソールでは WebSphere Application Server 環境の構成を表示および変更できます。また、Web アプリケーション (HATS アプリケーションなど) のインストール、開始、停止、およびアンインストールを実行できます。WebSphere コンソールのユーザー役割 (管理者、コンフィギュレーター、オペレーター、またはモニター) では、それぞれに WebSphere Application Server 管理コンソールで実行できる機能が異なります。これらのセキュリティーの役割およびポリシーについて詳しくは、以下の WebSphere Application Server Knowledge Center のリンクを参照してください。

- 管理ロールとネーミング・サービスの許可 (http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/csec_adminconsole.html)
- デフォルトの MBean セキュリティー・ポリシー (http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/cjmx_admin_defmbsec.html)

WebSphere グローバル・セキュリティーが有効な場合は、HATS 役割が設定されているユーザー ID を使用して、HATS アプリケーションで HATS 管理コンソールに接続でき、また特定の HATS アプリケーション内でのみ各管理操作を実行できます。他の HATS アプリケーションをリモート側で管理するために管理の有効範囲を変更しようとすると、WebSphere セキュリティーによりユーザー ID の権限が検査されます。ユーザー ID が、管理者権限またはオペレーター権限が設定された有効な WebSphere Application Server 管理コンソール・ユーザーでない場合は、JMX 呼び出しがブロックされ、HATS リモート管理が正しく機能しません。

例えば、ユーザー ID が WebSphere Application Server 管理コンソール・ユーザーとして定義されていないか、またはモニター権限のみが設定されている場合は、HATS 管理コンソールの「管理の有効範囲」の下の「アプリケーションの選択」リストに、他の HATS アプリケーションが表示されません。他のアプリケーションがリストされていないために、管理の有効範囲を変更できません。

もう 1 つの例として、コンフィギュレーター権限が設定された WebSphere Application Server 管理コンソール・ユーザーとして定義されているユーザー ID の場合、このリストには他の HATS アプリケーションが表示されるので、ユーザーが有効範囲を別のアプリケーションに変更できます。ただし、HATS 管理コンソールに表示される情報が正しくないことがあり、この場合リモート管理操作で試行した変更はすべて無効になります。

したがって、WebSphere グローバル・セキュリティーが有効である場合は、HATS 管理コンソールの使用に関して次の 2 つのオプションがあります。

オプション 1: HATS と WebSphere Application Server のユーザー役割を分離する
HATS 役割にマップされているユーザー ID に対して WebSphere Application Server 管理権限を付与しない場合は、すべての HATS アプリケーションに HATS 管理コンソールを組み込む必要があります。また、各 HATS アプリケーションを管理する際に、その HATS アプリケーションの HATS 管理コンソールを使用して別々に管理する必要があります。リモート管理は使用できません。

オプション 2: HATS と WebSphere Application Server のユーザー役割を結合する
単一の HATS アプリケーションで HATS 管理コンソールを使用し、管理の有効範囲を変更して他の HATS アプリケーションをリモートで管理するには、HATS 役割にマップされているユーザー ID を、WebSphere Application Server で管理者役割またはオペレーター役割が設定された WebSphere 管理コンソール・ユーザーとして定義する必要があります。これらのユーザーは、HATS 管理コンソールで管理の有効範囲を変更でき、マップされている特定の HATS 役割に許可されているその他の HATS 管理用タスクを実行できます。また、ユーザーは WebSphere Application Server 管理コンソールにログインして、マップされている WebSphere Application Server 管理コンソール・ユーザー役割に許可されている WebSphere 管理用タスクを実行できます。

いずれのオプションでも、WebSphere グローバル・セキュリティーが有効である場合は、「**アプリケーション・セキュリティーを有効にする (Enable application security)**」オプションが選択されていることを確認してください。 WebSphere Application Server V6.x の場合、WebSphere 管理コンソールから、「**セキュリティー**」>「**セキュアな管理、アプリケーション、およびインフラストラクチャー (Secure administration, applications, and infrastructure)**」>「**アプリケーション・セキュリティー**」>「**アプリケーション・セキュリティーを有効にする (Enable application security)**」を選択します。WebSphere Application Server V7.x および v8.x の場合、WebSphere 管理コンソールから、「**セキュリティー**」>「**グローバル・セキュリティー**」>「**アプリケーション・セキュリティー**」>「**アプリケーション・セキュリティーを有効にする (Enable application security)**」を選択します。

要約すると、WebSphere グローバル・セキュリティーが有効な場合は、HATS リモート管理が適切に機能するように、HATS 役割にマップされているユーザー ID を、WebSphere Application Server の管理者役割またはオペレーター役割にもマップする必要があります。

HATS 管理コンソールの役割

HATS 管理コンソールは WebSphere セキュリティーにバインドされています。つまり、WebSphere Application Server セキュリティーが使用可能な場合には、システム管理者には HATS 管理タスクを実行するための適切な認証が必要となります。
HATS により使用されるセキュリティー機能は、Java EE フォーム・ベースの認証プロセスに基づいています。WebSphere Application Server セキュリティーが使用可能な場合には、HATS 管理コンソールの操作はユーザー ID に定義されている役割に基づいて制限されます。HATS 管理コンソールの使用には、次の 3 つの役割が定義されています。各役割ではそれぞれ実行できる機能が異なります。

HATSAdministrator

この役割は完全な管理能力を備えており、以下のすべてのタスクを実行できます。

- ライセンス情報の管理
- 管理の有効範囲の選択
- アクティブな接続、接続プール、接続プール定義、ユーザー・リスト、およびログの表示
- アクティブな接続の終了
- ログおよびトレース・ファイルの表示
- ログおよびトレース・オプションの設定
- 端末の表示を確認することによる重要データである可能性のあるデータの表示、トレース設定の変更、既存の接続の終了、トレース・ファイルのダウンロード。

デフォルトでは、HATSAdministrator 役割は「すべての認証済み」ユーザーにマップされています。

HATSOperator

この役割は、重要データである可能性のあるデータにアクセスできない点を除いて役割 HATSAdministrator と同等です。この役割が設定されているユーザーは、端末の表示のトレースを使用可能にすることができません。また、トレース・ファイルもダウンロードできません。

HATSMonitor

この役割では、ユーザーが重要データ以外のデータ (アクティブ接続、接続プール、ユーザー・リスト、およびログなど) だけを表示できます。この役割が設定されているユーザーは、設定 (ライセンス管理設定、ログおよびトレース・ファイル・オプション、端末の表示の設定など) の変更と、トレース・ファイルのダウンロードを実行できません。

ユーザーまたはユーザー・グループへの役割のマッピングは、WebSphere 管理コンソールを使用して実行されます。

WebSphere Application Server V6.x の場合、適切な HATS 役割について、「**セキュリティー・ロールをユーザー/グループにマップ**」パネルで「**認証済みすべて**」オプションに必ずチェックマークを付けてください。WebSphere Application Server V7.x および V8.x の場合、適切な HATS 役割について、「**セキュリティー・ロールをユーザー/グループにマップ**」 パネルの「**特殊サブジェクトのマップ (Map Special**

Subjects)」ドロップダウンを使用して、「**アプリケーション・レルム内の認証済みすべて (All Authenticated in Application Realm)**」オプションが選択されていることを確認してください。

HATS 管理コンソールの開始

HATS 管理コンソールは、Web ベースです。 Web ブラウザーを使用して、HATS 管理コンソール・サーブレットである **HATSAdminServlet** により提供されるユーザー・インターフェースを介して管理タスクを実行します。

HATS 管理コンソールを始動するには、ブラウザーで URL **http://***host:port/appname/***hatsadmin/admin** をロードします。ここで、***host*** は使用するホスト名、***port*** は HATS アプリケーションが稼働しているアプリケーション・サーバーの HTTP トランスポートのポート番号、***appname*** は管理サポートが含まれている HATS アプリケーションの名前です。

注: URL 要求が WebSphere に対応して構成されている Web サーバーに送信される場合は、**ポート** の指定は不要です。

HATS 管理コンソールを始動すると、サーバーのデフォルト言語が使用されます。ただし、この言語は変更できます。言語を変更するには、ナビゲーション・フレームの「**プリファレンス**」を選択し、言語を選択してからブラウザーを再開始します。ブラウザーでは新たに選択した言語がデフォルトになります。この言語を変更するには、「**プリファレンス**」を選択して別の言語を選択します。

注:

1. HATS 管理コンソールにアクセスするブラウザーの Cookie が無効になっている場合は、次の変更が発生します。
   - HATS 管理コンソールでユーザーの言語設定が追跡されません。
   - ブラウザーの言語は、Accept-Language ヘッダーに基づいて、ブラウザーの優先言語により決定します。
   - 管理コンソール・プリファレンスに言語選択項目は表示されません。
2. サポート対象 Web ブラウザーのリスト、および制限については、「System Requirements for Rational Host Access Transformation Services」(http://www.ibm.com/support/docview.wss?uid=swg27011794) および「Rational Host Access Transformation Services 9.0 - Known issues and workarounds」(http://www.ibm.com/support/docview.wss?rs=3441&uid=swg27038663) を参照してください。

HATS Toolkit での管理コンソールの開始

HATS Toolkit では、「サーバーでデバッグ」または「サーバーで実行」モードで HATS 管理コンソールを開始できます。

特定のプロジェクトに対して HATS 管理コンソールを起動するには、「**HATS Projects**」ビューでプロジェクトを右クリックし、「**管理コンソールをオープン**」をクリックします。

プロジェクトがローカル・テスト環境でまだ実行されていない場合は、「サーバーでデバッグ」モードを使用してこのプロジェクトが起動され、Web ブラウザー・ページで管理コンソール URL が表示されます。

「サーバーで実行」モードでプロジェクトが既に実行されている場合は、サーバーがデバッグ・モードで実行されていないことを示すプロンプトが表示されます。**「現行モードで続行 (Continue in the current mode)」**オプションを選択すると、管理コンソールが「サーバーで実行」モードで開きます。

各種テスト・モードでの実行方法について詳しくは、*「スタートアップ・ガイド」*の『Web プロジェクトのテスト・モード』を参照してください。

HATS Toolkit で管理コンソールを実行するときに変更するランタイム設定は、使用しているモードによって異なります。詳しくは、450 ページの『問題判別コンポーネントの管理』を参照してください。

HATS 管理コンソールの機能の使用

このセクションでは、HATS 管理コンソールのさまざまな機能について説明します。

管理の有効範囲の選択

管理の有効範囲を選択できます。HATS 管理コンソールでは、HttpSession の存続期間にわたる管理の有効範囲のリストが維持されます。最初の管理の有効範囲は、デフォルトでは、HATS 管理コンソールへのアクセスに使用される HATS アプリケーションの HATS Enterprise Application (.ear) として選択されます。同じ WebSphere セル内に展開されている別の HATS Enterprise Application を管理するように管理の有効範囲を変更できます。

管理の有効範囲の変更で重要なことは、HATS 管理コンソールがインストールされている HATS Enterprise Application を検出できるようにホストとポートに関する情報を提供することです。指定されているポート値は、アクティブな WebSphere BOOTSTRAP_ADDRESS ポートでなければなりません。WebSphere Application Server、ノード、およびセルにはそれぞれ、BOOTSTRAP_ADDRESS ポートがあります。1 つの WebSphere リソースにポート値を指定すると、そのリソースに展開されている HATS Enterprise Application の管理が可能になります。WebSphere ノードのホストとポートを指定すると、そのノードにインストールされているすべての HATS Enterprise Application の管理が可能になります。BOOTSTRAP_ADDRESS ポートの詳細については、WebSphere Application Server の資料を参照してください。

機密保護機能のある環境では、管理者が管理の有効範囲を変更しようとすると、HATS 管理コンソールで次のメッセージが表示されることがあります。

選択された有効範囲の管理中に問題が発生しました。HATS アプリケーションがありません。管理の有効範囲は変更されませんでした。

機密保護機能のある WebSphere 環境で、管理対象の使用可能なアプリケーションのリストにデータを設定するための通信を実行する権限がユーザーにない場合に、この問題が発生することがあります。

WebSphere でユーザー・レジストリーにローカル・オペレーティング・システムを使用している場合、HATS 管理コンソールへのアクセスで使用するユーザー名に、ローカル・オペレーティング・システムのレジストリーでの管理者権限を設定する必要があります。WebSphere でユーザー・レジストリーに LDAP を使用している場合は、HATS 管理コンソールへのアクセスで使用するユーザー名は、WebSphere で LDAP へのアクセスで使用するサーバー ID である必要があります。

ライセンスの管理

左側のナビゲーション・フレームの「ライセンス管理」という見出しの下に、ライセンス情報を管理するための 2 つのパネルがあります。

- 「ライセンス使用」パネル: ノード内のすべてのアプリケーション・サーバーのライセンス使用状況を追跡します。HATS は、ホストまたはデータベース・リソースへの HATS 接続の数を追跡し、その値が購入したライセンスの数を超過すると、メッセージをログに記録します。
- 「ライセンス設定」パネル: 所有するライセンスのタイプ (**「許可ユーザー」**または**「バリュー・ユニット」**) を指定します。「許可ユーザー」ライセンスを所有している場合は、所有ライセンス数を指定します。有効な値は 1 から 50000 です。また、ライセンスのトラッキング情報を収集するかどうか、およびトラッキング情報の書き込み先ファイルのテンプレート名を指定します。デフォルトのテンプレート・ファイル名は license.txt です。ライセンス設定の指定について詳しくは、*「スタートアップ・ガイド」* の『HATS ランタイムおよびライセンス設定を有効にする』を参照してください。

接続のモニター

左側のナビゲーション・フレームの「接続管理」という見出しの下に、各種接続をモニターできる独立したパネルがあります。

- 「ホスト接続」パネル: 3270、5250、またはホスト上で稼働している VT アプリケーション (Telnet 経由でアクセスできる IBM i サーバーやその他のサーバーなど) への接続。
- 「データベース接続」パネル: HATS アプリケーションから JDBC (Java Database Connection) インターフェースを介したデータベースへの接続。

これらの表示画面に表示される各接続は、バックエンド・データ・ソース (3270 アプリケーションやデータベースなど) への通信リンクを示します。これらの表示画面では、データ・ソースとユーザーの接続と、ユーザーの接続 ID に関する詳細情報を確認できます。3270 接続用の LU またはプール名が存在する場合は、それも表示されます。また、セッションをシャットダウンすることもできます。

接続プールのモニター

左側のナビゲーション・フレームの「接続管理」という見出しの下にある「接続プール」パネルには、定義されている接続プールに関する情報が表示されます。接続プールの 1 つ以上の接続が割り振られるまでは、接続プールはリストされません。

接続プールは、3270 アプリケーションやデータベースのようなバックエンド・データ・ソースへの通信リンクの集合です。クライアント要求の代理として統合オブジェクトまたはアプリケーションが実行されると、このオブジェクトまたはアプリケーションはプールから使用可能な接続を取得し、この接続を使用してデータ・ソー

スへアクセスし、接続をプールに戻します。接続プールが使用可能になる場合、接続の確立におけるプロセッサー使用は、最初の接続の使用時に吸収されます。この接続を再利用すると、前回確立された接続が利用され、処理が高速になります。

プール定義のモニター

左側のナビゲーション・フレームの「接続管理」という見出しの下にある「接続プール定義」パネルには、ホストおよびデータベース接続のプール定義情報が表示されます。定義されているプールから 1 つ以上の接続が割り振られるまでは、プール定義はリストされません。

プール定義は、データ・ソース接続の集合に関する情報を提供します。プール定義には、関連する一部の定義 (接続とマクロの定義やユーザー・リスト名など) が関連付けられています。プール定義は、他のいくつかの定義間の関連を作成するほかに、最小および最大プール・サイズ (アクティブ接続数に換算) や接続タイマーなど、プール内のすべての接続に対する構成パラメーターを提供します。

ユーザー・リストとユーザー・リスト・メンバーのモニター

左側のナビゲーション・フレームの「ユーザー管理」という見出しの下にある「ユーザー・リスト」パネルには、ユーザー・リストに関する情報が表示されます。

ユーザー・リストは、特定の接続プール定義に関連付けられているユーザー ID やその他の接続固有パラメーターを示します。ユーザー ID ごとに 1 つの接続だけが許可されるシステムでは、1 つのプールでアクティブな接続の数はユーザー ID の数によって決まります。

問題判別コンポーネントの管理

HATS 管理コンソールでは、問題解決時に役立つロギングとトレースをモニターできます。左側のナビゲーション・フレームの「トラブルシューティング」という見出しの下にあるリンクを選択して、ログ内のメッセージを表示したり、ログ設定やトレース設定を変更したりできます。このような問題判別補助機能の設定は、プロパティー・ファイルに維持されています。

ランタイム環境で実行されている HATS 管理コンソールへのアクセス中に設定を変更すると、この変更は runtime.properties ファイルに保存されます。この方法は推奨しませんが、ファイルを直接編集して設定を変更することもできます。このファイルは HATS Enterprise Application の *was_dir*/installedApps/*ear_name* ディレクトリーに格納されています。

HATS Toolkit 内のローカル・テスト環境で実行されている HATS 管理コンソールへのアクセス中に設定を変更すると、この変更は異なる 2 つのファイルのいずれか 1 つに保存されます。「サーバーで実行」モードで実行している場合は、変更は runtime.properties ファイルに保存されます。「サーバーでデバッグ」モードで実行している場合は、変更は runtime-debug.properties ファイルに保存されます。詳しくは、447 ページの『HATS Toolkit での管理コンソールの開始』を参照してください。ファイルを直接編集して設定を変更することもできます。これらのファイルは HATS Toolkit の .ear プロジェクトのルートにあります。 HATS Toolkit で runtime.properties ファイルが変更された場合、その変更は保持され、HATS アプリ

ケーションをランタイム環境に展開するときに有効になります。
runtime-debug.properties ファイルは、ランタイム環境では無視されます。

注: WebSphere Application Server for z/OS® 上で実行している場合、runtime.properties ファイルを編集するときには ASCII エディターを使用してください。この場合、このファイルをワークステーションに例えば FTP を使用して転送し、ファイルを編集した後でこのファイルを送り返します。また、アプリケーション・サーバーのアクティブなサーバントごとに runtime.properties のコピーがあります (サーバントのアドレス・スペース ID により区別されます)。アプリケーション・サーバーの再始動後も変更を維持するには、基本 runtime.properties ファイルを変更します。

これらのファイルでのログ設定とトレース設定について詳しくは、557 ページの『付録 A. ランタイム・プロパティー・ファイル』を参照してください。

ログ・ファイルとトレース・ファイルの名前

runtime.properties に記述されている基本ログおよびトレース・ファイルの名前は、各アプリケーション・サーバーの一意のログ・ファイルおよびトレース・ファイルのセットを生成するときにテンプレートとして使用されます。デフォルトの基本トレース・ファイル名は *trace.txt*、デフォルトの基本ログ・ファイル名は *messages.txt* です。アプリケーション・サーバーのログ・ファイルおよびトレース・ファイルのテンプレートを生成するときには、基本ファイル名にアプリケーション・サーバー名が付加されます。このアプリケーション・サーバー名は、WebSphere で定義されているアプリケーション・サーバーの完全修飾名と同一です。この名前は *cell_node_server* です。

- *cell* は WebSphere セルの名前です。
- *node* は WebSphere ノードの名前です。
- *server* は WebSphere ノードの名前です。

HATS を実行しているアプリケーション・サーバーの名前は、アンダースコアー文字 (_)、アプリケーション・サーバー名、アンダースコアー文字 (_) をこの順序で連結したものになります。

例えばトレース・ファイルを使用する場合、この名前は *trace _cell_node_server_.txt* となります。さらに、複数のファイルを区別する場合、この名前に索引 (1、2、3 ...) を追加します。複数のトレース・ファイルが構成されている場合、このアプリケーション・サーバーのトレース・ファイル名は *trace_cell_node_server_1.txt, trace_cell_node_server_2.txt ...* となります。

注: WebSphere Application Server for z/OS 上で実行している場合、アプリケーション・サーバーは複数のサーバントを実行できます。この場合、各サーバントが固有のログ・ファイルとトレース・ファイルを生成します。これらのファイルを区別するため、名前のサーバー・コンポーネント部分にはサーバントのアドレス・スペース ID (ASID) が接頭部として追加されます (例: *trace_asid_1.txt*)。

ログの表示

「ログの表示」パネルを使用して、HATS ログ・ファイルを表示したり、ファイル全体をダウンロードしたり、ファイルをクリアしたりできます。ログ・ファイルには、次の 2 種類のイベントに関する情報が記録されます。

エラー・イベント
操作の完了を妨げている問題。一般にエラー・イベントでは、問題を訂正する何らかのアクションを実行する必要があります。

警告イベント
問題を訂正するアクションが必要な可能性のある予期しない状況の発生。警告イベントはエラー・イベントほど重大ではありません。

ログ・ファイルは、管理者がトラブルシューティングを行うときに参照し、参考情報として使用することを目的としています。ログに表示される多くのメッセージが、*「HATS メッセージ」* に記載されています。この資料では、必要に応じて実行できる問題訂正アクションを提案しています。

注: Windows では、HATS アプリケーションの実行中に標準出力ログと標準エラー・ログの名前の変更や削除はできません。トラブルシューティング手順で、ディスク・スペースの再利用や情報バンドル・ファイルのサイズの最小化のためにこれらのログの名前を変更するか、または削除する必要がある場合には、HATS を表すアプリケーション・サーバーを停止する必要があります。ただし、WebSphere ノードは引き続き実行できます。

ログ・オプションの設定

「ログ設定」パネルでは、ログ・ファイルへの情報と警告イベントの書き込みを制御し、ログ・イベントの書き込み先ファイルを定義できます。ログ・ファイル名の指定方法については、451 ページの『ログ・ファイルとトレース・ファイルの名前』を参照してください。ファイルがすでに存在している場合には、新しいイベントはこのファイルに追加されます。エラー・イベントは常にログ・ファイルに書き込まれます。維持するログ・ファイルの数と各ログ・ファイルの最大サイズも定義できます。

トレース・オプションの設定

「トレース設定」パネルを使用して、次のタイプのトレース・オプションを 1 つ以上選択できます。

ランタイムのトレースを使用可能にする
プロジェクト内で接続に関する HATS ランタイムのトレースを使用可能にします。これらのランタイム接続を異なるレベルでトレースする場合には、557 ページの『付録 A. ランタイム・プロパティー・ファイル』でトレースの設定に関する説明を参照してください。

ウィジェットのトレースを使用可能にする
プロジェクト内でウィジェットに関する HATS ランタイムのトレースを使用可能にします。ウィジェットを異なるレベルでトレースする場合には、557 ページの『付録 A. ランタイム・プロパティー・ファイル』でトレースの設定に関する説明を参照してください。

アクションのトレースを使用可能にする
プロジェクト内でイベント・アクションに関する HATS ランタイムのトレ

ースを使用可能にします。トレース・イベントを異なるレベルでトレースする場合には、 557 ページの『付録 A. ランタイム・プロパティー・ファイル』でトレースの設定に関する説明を参照してください。

コンポーネントのトレースを使用可能にする

プロジェクト内でコンポーネントに関する HATS ランタイムのトレースを使用可能にします。コンポーネントを異なるレベルでトレースする場合には、 557 ページの『付録 A. ランタイム・プロパティー・ファイル』でトレースの設定に関する説明を参照してください。

ユーティリティーのトレースを使用可能にする

プロジェクト内でランタイム・ユーティリティーに関する HATS ランタイムのトレースを使用可能にします。これらのランタイム・ユーティリティーを異なるレベルでトレースする場合には、 557 ページの『付録 A. ランタイム・プロパティー・ファイル』でトレースの設定に関する説明を参照してください。

統合オブジェクトのトレース

プロジェクト内の特定の統合オブジェクトのトレースを使用可能にします。クラス名指定フィールドに、1 つ以上のパターンを使用してトレースする統合オブジェクトを指定します。各パターンには、1 つ以上のワイルドカード (*) 文字を使用できます。例えば IntegrationObject.Callup* は、`Callup` で始まる名前を持つすべての統合オブジェクトに対してトレースを使用可能にすることを指定します。すべての統合オブジェクトをトレースするには、IntegrationObject.* を指定します。複数のパターンを指定する場合には、各パターンをコンマで区切ってください。HATS 管理コンソールでは、統合オブジェクトの設定が最大トレース・レベルに設定されます。統合オブジェクトを異なるレベルでトレースする場合には、 557 ページの『付録 A. ランタイム・プロパティー・ファイル』でトレースの設定に関する説明を参照してください。

アプレットのトレース

アプレットのトレースを使用可能にします。トレースのレベルを上げるほど、より多くのトレース情報が生成されます。アプレットのトレースをオンにすると、サーバー・サイドのトラブルシューティング・メッセージが出力されます。アプレットのトレース設定の詳細については、 557 ページの『付録 A. ランタイム・プロパティー・ファイル』を参照してください。

ユーザー・マクロのトレース

マクロ実行をトレースします。このトレースはシステム・パフォーマンスに影響するため、このトレースはマクロをデバッグする目的でのみ使用することをお勧めします。

端末の表示

サーバーでリアルタイムに実行されているあらゆる接続のホスト端末画面を表示できます。端末の表示を使用可能にすると、その後に作成された接続の端末設定のみが表示されます。端末の表示の詳細については、 454 ページの『テストとデバッグでの端末の表示の使用』を参照してください。

注: サーバーで実行時に端末の表示をオンにするかどうかを尋ねるダイアログを HATS で表示するかを決定できます。詳しくは、 149 ページの『HATS プリファレンスの使用』を参照してください。

ホスト・シミュレーション・トレースの記録

実行時にホスト・シミュレーションの記録を使用可能にするには、「**ホスト・シミュレーションの記録を使用可能にする**」ボックスを選択します。ホスト・セッションの次回開始時に記録が開始され、ホスト・セッションのクローズ時 (ユーザーがアプリケーション・キーパッドの切断ボタンをクリックする時点など) に記録が終了されます。トレースがアクティブなときにユーザーが Web ブラウザーを閉じると、HTTP セッションがタイムアウトになるまでトレースが続行されます。トレース・ファイルはサーバーのホスト・シミュレーション・フォルダー (例: <*WebSphere install root*>¥AppServer¥profiles¥AppSrv01¥installedApps¥<*Node ID*>¥<*ear file name*>¥<*ApplicationName*>.war¥WEB-INF¥profiles¥hostsimulations) に保存されます。トレース・ファイル名は、`ApplicationName_ConnectionName_Date(yyyymmdd)_Time(hhmmss)_Number` というテンプレートを使用して指定されます (例えば、MyApplication_main_20060101_134543_1)。

複数のホスト・セッションを同時にトレースするため、ホスト・シミュレーションでは複数の TCP/IP ポートが使用されます。「**記録の対象となる TCP/IP ポート範囲**」フィールドを使用してサーバーが実行時に使用するポート番号の範囲を設定します。デフォルトの範囲開始ポートは **7021**、デフォルトの終了ポートは **7050** です。

IBM サービス・トレース・オプション

HATS 管理コンソールの「**サービス・トレース・オプションの表示**」チェック・ボックスを使用して、トレースをオンにできます (IBM サービス担当員から指示された場合)。IBM サービス担当員に指示された場合以外は、これらのトレースは使用しないでください。

トレース出力

トレース・ファイルへのトレース情報の書き込みを制御し、ログ・イベントの書き込み先ファイルを定義できます。トレース・ファイル名の生成方法については、451 ページの『ログ・ファイルとトレース・ファイルの名前』を参照してください。ファイルがすでに存在している場合には、新しいトレース情報はこのファイルに追加されます。維持するトレース・ファイルの数と各トレース・ファイルの最大サイズも定義できます。

端末の表示機能

テストとデバッグでの端末の表示の使用

テスト・システムでアプリケーションをデバッグするときに、サーバーのディスプレイ上にホスト画面を表示する端末ウィンドウを作成するかどうかを制御できます。端末ウィンドウに表示されるホスト画面では、実行時に HATS アプリケーションとホスト・アプリケーションの間の対話を監視できます。また、端末ウィンドウのホスト画面を使用してホスト・アプリケーションと対話することもできます。このオプションがオンの間に確立された接続の端末画面のみが作成されます。既存の接続またはアイドル状態の接続には作成されません。ただし、HATS 管理コンソールの「ホスト接続」パネルには、選択されている接続の端末の表示をオンまたはオフにする「**表示状況の切り替え (Toggle Display Status)**」ボタンがあります。

注意:
「端末の表示」オプションをオンにすると、サーバーのパフォーマンスに深刻な影響を及ぼしたり、サーバーで過負荷が発生する可能性があります。多数の接続があるサーバーではこのオプションを使用しないでください。端末の表示は、テスト・システムでのアプリケーション開発時のデバッグ作業で使用することを目的としており、負荷の高い実動サーバーでの使用には適していません。

新しいホスト接続で端末の表示をオンにするには、左側のナビゲーション・フレームにある「**トラブルシューティング**」という見出しの下の「**トレース設定**」パネルの「**端末の表示を使用可能にする**」ボックスを選択します。ただし、「**トレース設定**」パネルの「**端末の表示を使用可能にする**」ボックスが選択されているかどうかに関係なくいつでも特定の接続の端末の表示をオンまたはオフにするには、「**ホスト接続**」パネルの「**端末の表示の切り替え**」ボタンを使用します。

注:

1. WebSphere Application Server が、Windows プラットフォームでサービスとして始動されるときに端末の表示が正しく行われるように、WebSphere Application Server のサービス設定で 「**デスクトップと対話することをサービスに許可する (Allow service to interact with desktop)**」 を選択する必要があります。
2. 表示端末を Windows プラットフォームで WebSphere Portal V7 と適切に連携させるには、ポータル・サーバーを Windows サービスではなく通常の Windows プロセスとして始動する必要があります。
3. IBM i プラットフォームで Native Abstract Windowing Toolkit (NAWT) を使用して表示端末機能に必要なグラフィックスを表示する方法については、必要に応じて、以下のいずれかの資料でセクション『グラフィカル・ユーザー・インターフェースを使用しないホスト上で Java アプリケーションを実行する』を参照してください。
   - *IBM Systems - iSeries プログラミング IBM Developer Kit for Java バージョン 5 リリース 4* (http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/rzaha/rzaha.pdf)
   - *System i プログラミング IBM Developer Kit for Java バージョン 6 リリース 1* (http://publib.boulder.ibm.com/infocenter/iseries/v6r1m0/topic/rzaha/rzaha.pdf)
   - *IBM i プログラミング IBM Developer Kit for Java 7.1* (http://publib.boulder.ibm.com/infocenter/iseries/v7r1m0/topic/rzaha/rzaha.pdf)
4. 他のプラットフォームで表示端末機能を構成する方法について詳しくは、HATS Knowledge Center においてセクション『IBM i, z Systems, and Unix platform-specific information』(http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0) を参照してください。

第 19 章 WebSphere Portal と HATS

IBM WebSphere Portal は、Web コンテンツおよびアプリケーションへの単一のアクセス・ポイントを提供するものであり、各ユーザーのニーズに合わせて個別に設定されます。

HATS を WebSphere Portal と統合することにより、HATS、WebSphere Portal、およびその他のポータル・アプリケーションの間でデータを自由にやり取りできます。WebSphere Portal の関連機能はすべて、組み込みツールや簡単な Java 開発によって HATS 内でも使用できます。WebSphere Portal 環境で HATS を実行した場合、HATS 機能の大部分は保持されます。詳細については、468 ページの『HATS ポートレットの考慮事項と制限』を参照してください。

HATS アプリケーションはポートレットとして WebSphere Portal 内で直接動作可能なため、WebSphere Portal と HATS は直接対話できます。標準の Java Portlet Specification API (JSR 168 または JSR 286、以降、標準ポートレット API と呼ぶ) に準拠した HATS ポートレットを作成できます。

注: 本書を通じて、標準ポートレット API (JSR 168 または JSR 286) に準拠する HATS ポートレットは、標準ポートレットと呼ばれます。 区別が必要な場合は、JSR 168 または JSR 286 が示されます。

HATS ビジネス・ロジック・クラスは WebSphere Portal API を使用でき、HATS JSP は WebSphere Portal JSP タグ・ライブラリーを組み込むことができます。ポートレットと拡張ポータル機能のプログラミングの詳細については、「*HATS Web アプリケーション・プログラマーズ・ガイド*」を参照してください。

HATS ポートレット・プロジェクトは、変換指向アプリケーションを対象としています。 WebSphere Portal で動作するようにモデル 1 統合オブジェクト・ベースのアプリケーションを適合させる方法については、「*HATS Web アプリケーション・プログラマーズ・ガイド*」を参照してください。 WebSphere Portal で実行する Struts 統合オブジェクト・ベースのアプリケーションを変換する方法については、ご使用の WebSphere Portal レベルの資料を http://www.ibm.com/developerworks/websphere/zones/portal/proddoc/index.html で参照し、「`Struts`」で検索を行います。

ポータルのツールの詳細およびヘルプを参照するには、Rational SDP ツールバーで「**ヘルプ**」メニューを選択し、「**ヘルプ目次**」を選択します。左側のヘルプ・リストの「**ポータルおよびポートレット・アプリケーションの開発 (Developing portal and portlet applications)**」をクリックして、ポートレット開発者用文書にアクセスします。

HATS ポートレットの作成

HATS ポートレットを作成する方法は 2 つあります。

- HATS の「プロジェクトの作成」ウィザードを使用し、サポートされる WebSphere Portal サーバーの 1 つを 「**ターゲット・サーバー**」とするよう指定することにより、新規ポートレットを作成できます。

© Copyright IBM Corp. 2003, 2015

- 既存の HATS Web プロジェクトからポートレットを生成する。この方法をとると、WebSphere Portal に HATS Web アプリケーションを移動する際に、そのアプリケーションの作成に費やした時間が無駄にならずに済みます。

新規 HATS ポートレット・プロジェクトの作成

新規 HATS ポートレット・プロジェクトを作成するには、次のいずれかの操作によって「プロジェクトの作成」ウィザードを起動します。

- メニュー・バーから「**HATS**」>「**新規**」>「**プロジェクト**」を選択します。
- メニュー・バーから「**ファイル**」>「**新規**」>「**HATS プロジェクト**」を選択します。
- ツールバーの「**HATS プロジェクトを作成**」アイコンをクリックします。

「プロジェクトの作成」ウィザードの「HATS プロジェクト」パネルで、以下のようにします。

1. ポートレット・プロジェクトの名前を入力します。
2. オプションで、説明を入力します。
3. デフォルトの場所をそのまま使用します。
4. 「**Web**」を選択して、これが Web プロジェクトであることを示します。

   注: Web デプロイメント・オプションが使用不可になっている場合、サーバー・ランタイムが定義されていないことを示しています。サーバー・ランタイムを定義するには、「**ウィンドウ**」>「**設定**」>「**サーバー**」>「**ランタイム環境 (Runtime Environments)**」と移動して、少なくとも 1 つのランタイム定義を追加します。
5. 「ターゲット・サーバー」フィールドで、以下のいずれかのターゲット WebSphere Portal サーバーを選択します。
   - **WebSphere Portal v7.0**
   - **WebSphere Portal v8.0**
6. 「エンタープライズ・アプリケーション・プロジェクト」フィールドに、ローカル・テスト環境でのテストに使用する EAR プロジェクトの名前を入力します。
7. 「ポートレット API」フィールドで、このポートレットが準拠する API として、「**JSR 286 ポートレット**」または「**JSR 168 ポートレット**」を選択します。
8. 管理コンソール用のサポート・ファイルを組み込むかどうかを選択します。

   注: このオプションは、HATS 標準ポートレットではサポートされていません。
9. 「**次へ**」をクリックして、プロジェクトの作成を通常どおりに続行します。

これにより、ローカル・テスト環境でのテストに使用可能な EAR プロジェクトを含む単一の HATS ポートレット・プロジェクトが生成されます。

HATS Web プロジェクトからのポートレットの生成

既存の HATS Web プロジェクトをポートレットとして実行できるようにするには、ポートレットの仕様に合うように HATS プロジェクトの構造と内容を変更する必要があります。HATS Toolkit では、通常の HATS Web プロジェクトをポートレ

ットに変換するプロセスが自動化されています。Rational SDP のポータル・ツール機能は、この変換プロセスの前提条件です。

既存の HATS Web プロジェクトから HATS ポートレット・プロジェクトを生成するには、「**HATS Projects**」ビューで Web プロジェクトを右クリックし、「**ポートレット・プロジェクトの生成**」を選択します。

「ポートレット・プロジェクトの生成」ウィザードの「ポートレット・プロジェクト生成オプション」パネルで、以下のようにします。

1. ポートレット・プロジェクトの名前を入力します。
2. 「ターゲット・サーバー」フィールドで、以下のいずれかのターゲット WebSphere Portal サーバーを選択します。
   - **WebSphere Portal v7.0**
   - **WebSphere Portal v8.0**

   注: HATS Web プロジェクトから HATS ポートレット・プロジェクトを生成する場合は、Web プロジェクトのターゲット・サーバーによってポートレット・プロジェクトのターゲット・サーバー・オプションが決まります。

   Web プロジェクトのターゲット・サーバーが WebSphere Application Server v7.0 の場合、生成されるポートレットのターゲット・サーバー・オプションは以下のとおりです。
   - **WebSphere Portal v7.0**
   - **WebSphere Portal v8.0**

   Web プロジェクトのターゲット・サーバーが WebSphere Application Server v8.0 または v8.5 の場合、生成されるポートレットのターゲット・サーバー・オプションは以下のとおりです。
   - **WebSphere Portal v8.0**
3. 「エンタープライズ・アプリケーション・プロジェクト」フィールドに、ローカル・テスト環境でのテストに使用する EAR プロジェクトの名前を入力します。
4. 「ポートレット API」フィールドで、このポートレットが準拠する API として、「**JSR 286 ポートレット**」または「**JSR 168 ポートレット**」を選択します。
5. 自動変換するファイルのタイプをこのリストから選択します。
   - 変換
   - 変換フラグメント
   - テンプレート
   - マクロ・ハンドラー
   - カスケーディング・スタイル・シート
6. 「**完了**」をクリックします。

選択した Web プロジェクトから HATS ポートレット・プロジェクトが作成されます。ポートレットが作成されたら、その他のカスタマイズを実行できます。HATS プロジェクトの変更の詳細については、107 ページの『第 5 章 HATS プロジェクトの変更』を参照してください。

注:

1. HATS Web プロジェクトを HATS ポートレット・プロジェクトに変換する場合、Web プロジェクトに対して構成されているプラグインがポートレット・プロジェクトに対してサポートされない場合は、Web 高速ログオン (WEL) 構成ファイル (hatswelcfg.xml) の手動での変更が必要になる可能性があります。WEL 構成ファイル (hatswelcfg.xml) が適切に更新されるようにするために、WEL 構成ユーザー・インターフェースを使用してサポートされるプラグインを追加することをお勧めします。
2. HATS Web プロジェクトから HATS ポートレット・プロジェクトを生成したあとに、あるプロジェクトに変更が加えられても、その変更はほかのプロジェクトには反映されません。HATS Web プロジェクトに加えた変更を反映させるには、変更した HATS Web プロジェクトから新規 HATS ポートレット・プロジェクトを生成します。
3. HATS ポートレット・プロジェクトを HATS Web プロジェクトに変換しなおすことはできません。
4. 既存の HATS V4、V5、V6、V7.0、V7.1、V7.5、V8.0、または V8.5 アプリケーションがある場合は、ポートレット・プロジェクトを生成する前に、そのアプリケーションを HATS V9.5 にマイグレーションする必要があります。 HATS V9.5 へのマイグレーションについて詳しくは、 17 ページの『第 2 章 HATS V9.5 へのマイグレーション』を参照してください。

HATS ポートレットの処理

HATS ポートレット・プロジェクトが HATS プロジェクトであることに変わりはありません。つまり、すべての HATS ウィザードおよびエディターをこのプロジェクト上で引き続き使用することができます。また、HATS ポートレット・プロジェクトもポートレット・プロジェクトです。つまり、プロジェクトですべての Portal Tools のエディターおよびランタイム構成が使用できるということです。HATS ポートレット・プロジェクトは、異なるアイコン でツリーに表示されます。

接続パラメーターおよびグローバル変数のオーバーライド

ポートレットの接続パラメーターおよびグローバル変数は、プロジェクト設定をカスタマイズすることによってオーバーライドできます。詳しくは、 130 ページの『接続パラメーターのオーバーライド』および 141 ページの『グローバル変数のオーバーライド』を参照してください。

標準ポートレット

HATS 標準 ポートレットでプロジェクト設定のオーバーライドを許可した場合、ユーザーは、ツールバーのポートレット・メニューから「**パーソナライズ (Personalize)**」を選択し、編集モードに切り替えることにより、接続パラメーターおよびグローバル変数をオーバーライドすることができます。

注: HATS は、WebSphere Application Server でのポートレット編集モードはサポートしていません。ポートレット編集モードを使用する代わりに、WebSphere Application Server 上で直接稼働するように HATS 標準ポートレットをデプロイ

すると、ユーザーは、ポートレットへのアクセスに使用する URL 内の接続パラメーターおよびグローバル変数をオーバーライドすることができます。

ポートレット編集モードに切り替わると、ユーザーに対して設定ページが表示されます。このページで、ユーザーは接続パラメーターおよびグローバル変数の設定をオーバーライドできます。これらの機能は、アプリケーションへの初回アクセス時に URL を使用して設定をオーバーライドすることをユーザーに許可する、HATS の Web アプリケーション用機能と同等のものです。

この設定ページの接続パラメーター・セクションにある接続パラメーターのドロップダウンには、プロジェクト設定でのオーバーライドを許可する Host On-Demand セッション・パラメーターがリストされます。ユーザーがオーバーライドを設定するには、このドロップダウン・リストからパラメーターを選択して「**値**」フィールドに値を入力し、「**追加**」をクリックします。デフォルトでは、最初に追加されるときにオーバーライド設定が使用可能になります。「**使用可能**」ドロップダウンから「**false**」を選択すると、この設定を無効にできます。オーバーライドを削除するには、パラメーター名の横にあるチェック・ボックスを選択して、「**削除**」をクリックします。

グローバル変数セクションでユーザーがオーバーライドを設定するには、グローバル変数の名前および値を入力して、「**追加**」をクリックします。オーバーライドの追加後に「**共用**」ドロップダウンから「**true**」を選択すると、そのグローバル変数が共用グローバル変数として識別されるようになります。デフォルトでは、最初に追加されるときにオーバーライド設定が使用可能になります。「**使用可能**」ドロップダウンから「**false**」を選択すると、この設定を無効にできます。オーバーライドを削除するには、グローバル変数名の横にあるチェック・ボックスを選択して、「**削除**」をクリックします。

ユーザーが「**保存して終了**」をクリックすると、ポートレット設定に設定が保存されます。

注:

1. ユーザーが入力した値は検証されません。
2. 新しい値を有効にするには、ポートレット・アプリケーションを切断してから再始動する必要があります。

ユーザーによるオーバーライド設定を許可するほかに、必要に応じて HATS JSR 286 または JSR 168 入力ポートレットを拡張することによって、接続パラメーターおよびグローバル変数のオーバーライドをプログラマチックに指定することができます。詳しくは、「*HATS Web アプリケーション・プログラマーズ・ガイド*」のセクション『Entry ポートレットの拡張』を参照してください。

注: こちらが拡張ポートレット・クラスを使用して接続パラメーターまたはグローバル変数のオーバーライドを指定した際に、ユーザーがポートレット編集モードを使用して同じオーバーライドを指定した場合は、こちらが拡張ポートレット・クラスで指定した値が、ユーザーが編集モードで指定した値によってオーバーライドされます。

.

ポートレット通信

WebSphere Portal は、ポートレットが情報を交換または共用するための複数の方法をサポートします。詳しくは、http://www.ibm.com/developerworks/websphere/zones/portal/proddoc/index.html において、ご使用の WebSphere Portal レベルの資料を参照して、「Portlet communication」で検索を行ってください。

このセクションでは、WebSphere Portal ポートレット通信の HATS サポートについて説明します。

JSR 168 ポートレット

HATS JSR 168 ポートレット・サポートは、WebSphere Portal 連携ポートレット・メカニズムを使用してポートレット通信ソリューションを提供します。

JSR 168 ポートレットは、プロパティーを使用して相互に通信します。 HATS JSR 168 ポートレットは、その他の JSR 168 ポートレット (HATS または非 HATS) との間でプロパティーの送受信を行うための機能をサポートします。通信するためには、ポートレットをワイヤリングする必要があります。

HATS JSR 168 ポートレットからのプロパティーの送信:

1. 送信するプロパティーの値を含むグローバル変数を作成します。これは、「**グローバル変数を抽出**」アクションや「**グローバル変数を設定**」アクションの実行など、いくつかの方法で行うことができます。JSR 168 ポートレット通信では、ゼロ個以上のストリングを含む (それ以外のタイプは含まない) ことができるグローバル変数のみを使用できます。グローバル変数は、JSR 168 プロパティーの値のラッパーと考えることができます。
2. HATS 画面イベントまたは一致しない画面アプリケーション・イベントに「**グローバル変数の送信**」アクションを追加して、グローバル変数の値をプロパティーとして送信します。これを行う方法については、 187 ページの『「グローバル変数を送信」アクション』を参照してください。

   このアクションを「**グローバル変数を抽出**」アクションとともに使用することによって、ユーザーが入力したデータを、あるポートレットから別のポートレットに送信することができます。例えば、「**変換を適用**」アクションと「**グローバル変数の送信**」アクションの間に「**グローバル変数を抽出**」アクションを追加することで、ユーザーが入力したデータを取り込み、それを他のポートレットに送信することができます。
3. HATS JSR 168 ポートレット・プロジェクトが実行されているときに「**グローバル変数の送信**」アクションを実行すると、指定されたグローバル変数のプロパティー名とストリング内容が WebSphere Portal に渡され、そこからこの HATS JSR 168 ポートレットにワイヤリングされているポートレットにプロパティーを受信するように通知します。

HATS JSR 168 ポートレットでのプロパティーの受信:

1. ポートレット設定を使用して、プロパティー名および、このポートレットで受信するプロパティーの値を含む、対応するグローバル変数を定義します。ポートレット設定には、プロジェクト設定エディターで「**その他**」タブをクリックしてから「**ポートレット**」をクリックすることによりアクセスします。これを行う方法の詳細については、 148 ページの『ポートレットの設定』を参照してください。

2. WebSphere Portal がソース・ポートレットからプロパティーを受信すると、ターゲット・ポートレットに通知されます。HATS JSR 168 ポートレット・ランタイム・サービスは、WebSphere Portal からプロパティー値を取得し、指定されたグローバル変数に格納します。繰り返しますが、JSR 168 通信では、ストリング (複数可) を含むグローバル変数のみを使用できます。

非 HATS JSR 168 ポートレットとの通信: Rational SDP には、連携ポートレットを実装するための GUI サポートがあります。 HATS JSR 168 ポートレットと通信するように非 HATS JSR 168 ポートレットを構成する場合は、以下のステップを実行します。

1. **「ウィンドウ」>「ビューの表示」>「その他」>「一般」>「プロジェクト・エクスプローラー」**と選択して、Rational SDP メニュー・バーから「プロジェクト・エクスプローラー」ビューを開きます。
2. 「プロジェクト・エクスプローラー」ビューで、ポートレット・プロジェクトの下にある**「ポートレット・デプロイメント記述子」**ノードを右クリックします。
3. **「連携 (Cooperative)」>「このポートレットによるデータ (ソース) の送信を使用可能にする」**、または**「連携 (Cooperative)」>「このポートレットによるデータ (ターゲット) の受信を使用可能にする」**と選択し、選択したポートレットの連携ポートレット・サポートを有効にするためのウィザードを開始します。
4. このウィザードでは、以下の項目を正しく設定してください。
   - データ型 URI = `http://com.ibm.hats.portlet.pb#<property_name>Type`。この URI では、名前空間の後にプロパティーのタイプが続きます。 HATS JSR 168 ポートレットで送信または受信するプロパティーは、名前空間として `com.ibm.hats.portlet.pb` を使用します。

     注: `Type` というワードは、大文字の `T` で始める必要があります。
   - Java タイプ = `java.lang.String`。 HATS ポートレットでは、プロパティーをストリングとして送信または受信します。

データ型 URI の編集: 「グローバル変数の送信」アクションまたはプロパティーを受信するためのポートレット設定を構成するときに、HATS は、そのプロパティーのデータ型 URI をデフォルトで `http://com.ibm.hats.portlet.pb#<property_name>Type` に設定します。 この URI を使用するのは、ソース・ポートレットとターゲット・ポートレットをワイヤリングする場合のみです。 WebSphere Portal でプロパティーを送信および受信するには、ソース・ポートレットとターゲット・ポートレットで同じ URI を使用する必要があります。何らかの理由でこの URI をデフォルト以外の値に構成する必要がある場合は、Portal のツールキットで以下のステップを実行することにより、プロパティーの URI を変更することができます。

1. **「ウィンドウ」>「ビューの表示」>「その他」>「一般」>「プロジェクト・エクスプローラー」**と選択して、Rational SDP メニュー・バーから「プロジェクト・エクスプローラー」ビューを開きます。
2. 「プロジェクト・エクスプローラー」ビューでプロジェクトを展開し、変更の対象となるプロパティーを表示します。プロパティーは、**「ポートレット・デプロイメント記述子」**ノードの下にあります。
3. プロパティーを右クリックし、**「データ型の編集」**を選択します。

4. 「連携ソースの編集」ウィザードまたは「連携ターゲットの編集」ウィザードで、「**データ型 URI**」を変更します。

JSR 286 ポートレット

HATS JSR 286 ポートレット・サポートは、WebSphere Portal ポートレット・イベント・メカニズムを使用してポートレット通信ソリューションを提供します。

JSR 286 ポートレットは、イベントを使用して相互に通信します。イベントは、名前と値タイプで定義されます。単一ポートレット内のイベント名は固有でなければなりません。 ポートレットは、それがイベントの公開 (送信) または処理 (受信) のどちらを行うかを宣言します。

HATS JSR 286 ポートレットからのイベントの送信:

1. 送信する 1 つまたは複数のオブジェクトを含むグローバル変数を作成します。これは、「**グローバル変数を抽出**」アクションや「**グローバル変数を設定**」アクションの実行など、いくつかの方法で行うことができます。1 つ以上の順序付け可能なオブジェクトを含む任意の HATS グローバル変数を通信で使用できます。グローバル変数は、JSR 286 イベントのオブジェクトのラッパーと考えることができます。
2. HATS 画面イベントまたは一致しない画面アプリケーション・イベントに「**グローバル変数の送信**」アクションを追加して、グローバル変数のオブジェクトをポートレット・イベントとして送信します。これを行う方法については、187 ページの『「グローバル変数を送信」アクション』を参照してください。

   このアクションを「**グローバル変数を抽出**」アクションとともに使用することによって、ユーザーが入力したデータを、あるポートレットから別のポートレットに送信することができます。例えば、「**変換を適用**」アクションと「**グローバル変数の送信**」アクションの間に「**グローバル変数を抽出**」アクションを追加することで、ユーザーが入力したデータを取り込み、それを他のポートレットに送信することができます。
3. HATS JSR 286 ポートレット・プロジェクトが実行されているときに「**グローバル変数の送信**」アクションを実行すると、指定されたグローバル変数のイベント名と内容が WebSphere Portal に渡され、そこからこの HATS JSR 286 ポートレットにワイヤリングされているポートレットにイベントを受信するように通知します。

HATS JSR 286 ポートレットでのイベントの受信:

1. ポートレット設定を使用して、このポートレットで受信できる、イベント名、イベントのオブジェクトを含める対応するグローバル変数、およびオブジェクト・タイプを定義します。ポートレット設定には、プロジェクト設定エディターで「**その他**」タブをクリックしてから「**ポートレット**」をクリックすることによりアクセスします。これを行う方法の詳細については、148 ページの『ポートレットの設定』を参照してください。
2. WebSphere Portal がソース・ポートレットからイベントのオブジェクトを受信すると、ターゲット・ポートレットに通知されます。HATS JSR 286 ポートレット・ランタイム・サービスは、WebSphere Portal からイベントのオブジェクトを取得し、指定されたグローバル変数に格納します。

非 HATS JSR 286 ポートレットとの通信: Rational SDP は、JSR 286 ポートレット・イベントを実装するための GUI サポートを提供します。 HATS JSR 286 ポートレットと通信するように非 HATS JSR 286 ポートレットを構成する場合は、以下のステップを実行します。

1. **「ウィンドウ」>「ビューの表示」>「その他」>「一般」>「プロジェクト・エクスプローラー」**と選択して、Rational SDP メニュー・バーから「プロジェクト・エクスプローラー」ビューを開きます。
2. 「プロジェクト・エクスプローラー」ビューで、ポートレット・プロジェクトの下にある**「ポートレット・デプロイメント記述子」**ノードを右クリックします。
3. **「イベント」>「このポートレットによるイベントの発行を使用可能にする (Enable this Portlet to Publish Event)」**または**「イベント」>「このポートレットによるイベントの処理を使用可能にする (Enable this Portlet to Process Event)」**を選択し、選択したポートレットに対して JSR 286 ポートレット・イベント・サポートを有効にするためのウィザードを開始します。
4. ポートレットの公開 (送信) と処理 (受信) の間でイベント名と値タイプが一致する必要があることに注意して、ウィザードを実行します。

通信のためのポートレットのワイヤリング

ポートレットのワイヤリングは、JSR 286 ポートレットのイベントまたは WebSphere Portal 連携ポートレット API を使用して通信するポートレット間の情報フローを誘導するために使用されます。ポートレットのワイヤリングとは、ポートレット・イベントまたは連携ポートレット・プロパティーの送受信を目的として、ポータル・サーバーのワイヤリング・ツールを使用して 2 つのポートレットを接続することです。ポートレットは、単一のイベント (またはプロパティー) についてのみワイヤリングされ、両方のポートレットがこのイベント (またはプロパティー) に同じ名前を指定する必要があります。ただし、関連付けられたグローバル変数の名前は異なっていてもかまいません。言い換えると、2 つのポートレットが 2 つのイベント (またはプロパティー) を送受信する場合、それらを接続する 2 つのワイヤー (それぞれのイベント (またはプロパティー) に対して 1 つ) が必要です。ワイヤリング手順を実行しないと、データは伝送されません。

カスタム PDT およびカスタム・テーブル

カスタム・プリンター定義テーブル (PDT) またはカスタム・テーブルを含むポートレット・プロジェクトをポータル・サーバーにデプロイする前に、ファイルまたはフォルダーを HATS ポートレット・プロジェクトの `¥Web Content` ディレクトリーにコピーする必要があります。各追加のタイプのファイルまたはフォルダーを以下に示します。

- カスタム・プリンター定義テーブル (PDT): コンパイルされた PDT ファイル、またはコンパイルされた PDT ファイルを含むフォルダー
- カスタム・テーブル: カスタム・テーブルの定義を含む customtable フォルダー

Web Services for Remote Portlets

Web Services for Remote Portlets (WSRP) は、ポータルで、ポートレット、アプリケーション、およびコンテンツを WSRP サービスとして提供する際に使用可能な規格です。他のポータルは、ポータル・ユーザーのリモート・ポートレットとして、これらの WSRP サービスを統合できます。標準ポートレットおよび IBM HATS ポ

ートレットを、他のポータルで消費可能なリモート・ポートレットとして指定できます。詳しくは、http://www.ibm.com/developerworks/websphere/zones/portal/proddoc/index.html において、ご使用の WebSphere Portal レベルの資料を参照して、「Using WSRP services」で検索を行ってください。

HATS ポートレットのテスト

Rational SDP 内に WebSphere Portal テスト環境が構成されている場合は、その環境を使用して、HATS 標準ポートレットを実動ポータル・サーバーに展開する前にテストすることができます。

WebSphere Application Server ローカル・テスト環境を使用して HATS 標準ポートレットをテストすることもできます。

HATS ポートレット・プロジェクトをテストする場合は、HATS Web プロジェクトのテストと同じ手順を使用します。詳しくは、107 ページの『概要』を参照してください。

HATS ポートレットのエクスポート

HATS ポートレットをデプロイメントのためにエクスポートするには、「HATS Projects」ビューでポートレット・プロジェクトを右クリックして、「**プロジェクトのエクスポート**」を選択します。エクスポートする .war ファイルのターゲット・ファイル名を 「**宛先**」ドロップダウンで入力し、「**完了**」をクリックします。.war ファイルのみがエクスポートされます。.ear ファイルを使用するのは、ローカル・テスト環境でのテストの場合のみです。

HATS ポートレットの .war ファイルをエクスポートしたら、WebSphere Portal の管理ツールを使用して、それを WebSphere Portal にインストールします。

標準ポートレットを WebSphere Application Server にインストールする場合は、WebSphere Application Server 管理コンソールを使用し、以下の説明を考慮に入れてください。

- インストール中に、コンテキスト・ルート名を指定する必要があります。このコンテキスト・ルート名は、インストール完了後にポートレット .war ファイルを開始するために必要です。インストール中に、.war ファイルを server1、または定義済み Web サーバー・インスタンスにデプロイします。
- ポートレット .war ファイルのインストール後、「**アプリケーション**」を展開して「**エンタープライズ・アプリケーション**」をクリックすることにより、WebSphere Application Server 管理コンソールを使用してコンテキスト・ルート名とポートレット名を確認できます。ご使用の .war ファイルをクリックし、「**Web モジュールのコンテキスト・ルート**」をクリックします。 Web モジュールと ContextRoot の値をメモします。これらの値は、HATS アプリケーションを開始するために使用されます。
- HATS アプリケーションを実行するには、ブラウザーに URL を入力します。例えば、`http://host:port/context_root_name/portlet_name` と入力します。ここで、`host` と `port` は、ポートレットをインストールした WebSphere Application Server ホストのホスト名とポートを表します。`context_root_name` には、ポート

レット .war ファイルのインストール時にコンテキスト・ルートに指定した値を使用し、portlet_name には、Web モジュールに指定した値を使用します。

HATS ポートレットの管理

HATS ポートレットは、HATS 管理コンソールを使用して管理できます。 HATS 標準ポートレットは、Portal Server または WebSphere Application Server のどちらにインストールしたかには関係なく、独立型の HATS 管理アプリケーションを使用して管理できます。

独立型の HATS 管理アプリケーションの使用

HATS 標準ポートレットを独立型の HATS 管理アプリケーションを使用して管理するには、以下のステップに従います。

1. HATS Toolkit で「**ファイル**」>「**新規**」>「**プロジェクト**」>「**HATS**」>「**HATS 管理プロジェクト**」と選択して、独立型の HATS 管理プロジェクトを作成します。 HATS 管理プロジェクトは、HATS Toolkit の「**ナビゲーター**」タブにのみ表示されます。
2. その HATS 管理 .ear プロジェクトを右クリックし、「**エクスポート**」>「**Java EE**」>「**EAR**」を選択することによって、.ear プロジェクトを .ear ファイルにエクスポートします。
3. 独立型の HATS 管理アプリケーションの .ear ファイルを、WebSphere Application Server 管理コンソールを使用して WebSphere Application Server に展開します。

   注:

   a. .ear ファイルは、WebSphere Portal サーバーと同じ WebSphere Application Server システムにインストールしてください。標準 HATS ポートレットを WebSphere Application Server に直接インストールする場合は、.ear ファイルを同じサーバーにインストールしてください。

   b. モジュールは、WebSphere Portal Server ではなく、WebSphere Application Server にマップするようにしてください (例えば、server1)。

   c. WebSphere Application Server V6.x の場合、HATS 管理者役割について、「**セキュリティー・ロールをユーザー/グループにマップ**」パネルで「**認証済みすべて**」オプションに必ずチェックマークを付けてください。WebSphere Application Server V7.x および V8.x の場合、「**セキュリティー・ロールをユーザー/グループにマップ**」パネルの「**特殊サブジェクトのマップ (Map Special Subjects)**」ドロップダウンを使用して、「**アプリケーション・レルム内の認証済みすべて (All Authenticated in Application Realm)**」オプションが選択されていることを確認してください。
4. ポートレットの管理に使用するユーザー ID は、**管理者**権限を付与された WebSphere Application Server 管理コンソール・ユーザーとして定義されている必要があります。これを実行するには、以下のステップを実行します。

   a. WebSphere Application Server 管理コンソールで、「**ユーザーおよびグループ (Users and Groups)**」>「**管理ユーザーのロール (Administrative User Roles)**」>「**追加**」を選択します。

b. ポートレットの管理に使用するユーザー ID を入力し、このユーザー ID を**管理者**の役割にマップします。

注: 詳しくは、 444 ページの『HATS 管理コンソールおよび WebSphere セキュリティー』を参照してください。

5. WebSphere Application Server 管理コンソールを使用して HATS 管理アプリケーションを開始します。
6. 独立型の HATS 管理コンソールにアクセスします。例えば、`http://<washostname>/<hatsadminprojectname>/hatsadmin/admin` という URL を使用します。

注: HATS 管理コンソールにアクセスする際は、ポートレットにアクセスする際とは異なるブラウザーを使用する必要があります。

a. 「**スタートアップ・ガイド**」>「**管理の有効範囲**」と選択します。
b. 「**新規**」をクリックします。
c. 「**ホスト**」フィールドに、ターゲット WebSphere Application Server ホストと同じホスト名を入力します。
d. 「**ポート**」フィールドに、ターゲット・サーバーの BOOTSTRAP_ADDRESS ポートを入力します。ターゲット・サーバーの BOOTSTRAP_ADDRESS ポートを調べるには、WebSphere Application Server 管理コンソールにアクセスして、以下のステップに従います。
   1) 「**サーバー**」>「**アプリケーション・サーバー**」と選択します。
   2) ターゲット・サーバー (WebSphere Portal サーバーまたは WebSphere Application Server (例: server1、標準 HATS ポートレットが WebSphere Application Server に直接インストールされている場合) のいずれか) をクリックします。
   3) 「構成」タブの「通信 (Communications)」にある「**ポート (Ports)**」を展開します。
   4) ターゲット・サーバーの BOOTSTRAP_ADDRESS ポートをメモします。
e. 「**OK**」をクリックします。
f. 新しく追加したホストを選択します。
g. 「**上記で選択した有効範囲内のすべてのアプリケーションを管理**」は選択しないでください。
h. HATS ポートレットのリストが表示されます。 HATS 管理コンソールを使用して、このリストから 1 つを選択することにより、一度に 1 つのポートレット・アプリケーションを管理します。

HATS ポートレットの考慮事項と制限

一般的な考慮事項と制限

以下に、HATS 標準ポートレットを使用する際の考慮事項および制限をリストします。

- ポートレットは、Web アプリケーション・アーカイブ (.war ファイル) としてインストールする場合にのみ、実動環境にインストールできます。 HATS では

.war ファイルごとに 1 つのプロジェクトしかないため、HATS アプリケーションのセットを単一のポートレット・アプリケーションとしてエクスポートすることはできません。

- WebSphere Portal 内での認証には Cookie が必要であるため、WebSphere Portal 内の HATS は Cookie を有効にして稼働させる必要があります。
- HATS Web プロジェクトから HATS 標準ポートレット・プロジェクトにコピーされた JSP ファイル (テンプレート、変換、変換フラグメント、およびマクロ・ハンドラーなど) は、変換しない限り動作しません。これらの JSP ファイルを変換するには、「**ポータルの JSP の変換**」アクションを使用します。
- Rational SDP の「新規ポートレット」ウィザードに HATS オプションが表示されますが、これらのオプションを選択しても、HATS リソースのない空のポートレットが生成されるだけです。
- HATS IBM ポートレットと HATS 標準ポートレットを、互いに干渉し合うことなく同じポータル・ページ上で実行することはできません。IBM の要求パラメーター処理と標準の要求パラメーター処理の違いについて詳しくは、http://www.ibm.com/developerworks/websphere/zones/portal/proddoc/index.html において、ご使用の WebSphere Portal レベルの資料を参照して、「`Handling and visibility of request parameters in portlets`」で検索を行ってください。
- HATS ポートレットがその他のポートレット (HATS または非 HATS) と同じポータル・ページにある場合、HATS ポートレットを切断すると、同じポータル・ページ上のその他のポートレットとの対話で発生するページ最新表示の結果として、ポートレットは自動的に再始動されます。 HATS 標準ポートレットは、ポータル・ページのテーマが `PortalWeb2` に設定されている場合にこの動作を示します。
- 特定の HATS 事前定義テンプレートを使用する複数の HATS ポートレットが、同じポータル・ページで実行されている場合、テンプレートが互いに干渉し合うことがあります。この状況は、すべてのテンプレートが同じカスケーディング・スタイル・シート (CSS) ルール名を使用している場合に発生します。この影響を受けるのは、template.jsp ファイルの CSS ルール名のみです。これは、これらの名前に接頭部 `div#<portletName>` が付いていないためです。この制限を回避するには、以下のいずれかを実行します。
  - この制限のない以下のいずれかの HATS 事前定義テンプレートを使用します。
    - Finance
    - Industry
    - Medical
    - Research
    - Transport
  - template.jsp ファイルを編集し、接頭部 `div#<portletName>` を各 CSS ルールに追加します。例えば、template.jsp ファイルに次のように記載されているとします。

```
A.BOTTOMBAR:visited,
A.BOTTOMBAR:link,
A.BOTTOMBAR:active {color: #55839A; font-size: 1em;}
```

```
table.ApplicationKeypad {
 color: blue;
 white-space: nowrap;
}
```

ポートレット・プロジェクトの名前が hatsPort914 であると仮定して、この例を以下のように変更します。

```
div#hatsport914 A.BOTTOMBAR:visited,
div#hatsport914 A.BOTTOMBAR:link,
div#hatsport914 A.BOTTOMBAR:active {color: #55839A; font-size: 1em;}

div#hatsport914 table.ApplicationKeypad {
 color: blue;
 white-space: nowrap;
}
```

- 以下の HATS 機能、アクション、および設定は、HATS ポートレットではサポートされていません。
  - 変換とテンプレートの編集時における Page Designer の「プレビュー」機能。
  - Dojo ウィジェット。
  - 「**HTTP 圧縮を有効にする**」設定。
  - 「**非同期更新**」アプレット。
  - 「**disconnectOnClose**」接続パラメーター。
  - 画面組み合わせのオプション「**動的キャッシュ・コンテンツ・ロードを使用する**」。
  - WebFacing アプリケーションとの HATS の相互運用性。
  - HATS 標準ポートレットは、WebSphere Portal または WebSphere Application Server プラットフォーム上でのみ実行することができます。
  - WebSphere Portal 信任状ボールト・プラグインを使用して HATS 標準ポートレットに Web 高速ログオン (WEL) サポートを追加すると、サポートされる WebSphere Portal サーバーで実行されます。ただし、信任状ボールト・サービス・サポートのない WebSphere Application Server で直接実行している場合、ホスト・システムへのログオンは失敗します。
  - 以下の HATS アクション、設定、および機能は、HATS 標準ポートレットではサポートされていません。
    - ポートレット・プロジェクトに HATS 管理コンソール・サポートを追加する機能。
    - 「新規ビジネス・ロジック」ウィザードで、認証データへのアクセス、認証データの取得、ポートレット・メッセージング、またはプロパティー・ブローカーの使用などの機能のサンプル・コードを生成する機能。
    - 「ポートレットのプレプレゼンテーション (Portlet Prepresentation)」(「ブロック区切り文字を追加」) アクション。
  - 「グローバル変数の送信」アクションを使用する際、「変換を適用」アクションと「グローバル変数の送信」アクションの間でサポートされるアクションは、以下のアクションだけです。
    - ビジネス・ロジックを実行
    - グローバル変数を抽出
    - データを挿入

- グローバル変数を設定
- グローバル変数を除去
- マクロ・トランザクションを実行
- 一時停止

– 統合オブジェクトを使用する際は、以下のことが適用されます。
  - モデル 1 Web ページのみがサポートされます。 Struts および JSF の Web ページはサポートされません。
  - HATS ポートレット・プロジェクトをコピーまたは名前変更すると、独立型の統合オブジェクトまたはチェーン内の最初の統合オブジェクトから生成されたモデル 1 JSP は、誤った接続仕様を参照することになります。以下の記述を変更する必要があります。

    ```
    SignOn.setHPubStartPoolName("<new_project_name>/<default_connection>");
    ```

  - Web プロジェクトからポートレット・プロジェクトを生成した際、以前に統合オブジェクトから生成したモデル 1 JSP ページに変更はありません。それらのページを再生成するか、手動で変更する必要があります。
  - グローバル変数およびテンプレートはサポートされていません。
  - 標準ポートレットで Web 高速ログオンを使用するように構成されている統合オブジェクトを実行するには、processRequest() メソッドを実行する前に setHPubPortletRequest(javax.portlet.PortletRequest) を呼び出して、その統合オブジェクトで PortletRequest を使用できるようにする必要があります。詳しくは、「*HATS Web アプリケーション・プログラマーズ・ガイド*」の『Web 高速ログオンの使用』を参照してください。
– HATS 管理コンソールを使用してアクティブなホスト接続を表示する際、HATS 標準ポートレットのクライアント接続に、ブランクの IP アドレスが表示されます。

第 20 章 WebFacing と HATS

WebFacing テクノロジーは、IBM i データ記述仕様 (DDS) ディスプレイ・ファイル・ソース・メンバーを既存の 5250 プログラムの Web ベースのユーザー・インターフェースに変換する機能を提供します。

WebFacing テクノロジーはインストール可能な HATS フィーチャーによって提供されています。IBM i フィーチャー用の IBM WebFacing Tool は、HATS インストール・プロセスの一部としてインストールできます。

HATS Toolkit の IBM WebFacing Tool for IBM i フィーチャーによって、 IBM i データ記述仕様 (DDS) およびユーザー・インターフェース・マネージャー (UIM) ヘルプ・パネル・ソース・メンバーを既存の 5250 プログラム用の Web ベース・ユーザー・インターフェースに変換する機能が提供されます。

WebFacing フィーチャーのインストールの詳細については、*「HATS スタートアップ・ガイド」*のセクション『HATS Toolkit のインストール』を参照してください。WebFacing 機能について詳しくは、『HATS アプリケーションの開発 (Developing HATS Applications)』トピックの下の製品ヘルプの『WebFacing アプリケーションの開発』トピックを参照してください。また、WebFacing 情報は http://www.ibm.com/support/knowledgecenter/SSXKAY_9.5.0 の HATS Knowledge Center でも使用可能です。

WebFacing との HATS 対話

HATS には、WebFacing テクノロジーと対話する方法が 2 つあります。1 番目の方法を使用すると、HATS Web アプリケーションは、Telnet 5250 サーバーへの接続の代わりに WebFacing サーバーへの接続を使用できます。2 番目の方法を使用すると、HATS Web アプリケーションと WebFacing アプリケーションをリンクして、WebFacing サーバーへの単一の接続を使用できる単一のエンタープライズ・アプリケーションとして相互に運用できます。それぞれの方法について、以下に説明します。

WebFacing サーバーへの HATS 接続

HATS には、5250 ホスト画面アプリケーションへ接続するための接続オプションが、以下のように 2 つあります。

- 5250 (Telnet)
- 5250W (WebFacing)

HATS と Telnet 5250 サーバー間のデータ・フローまたは HATS と WebFacing サーバー間のデータ・フローに差はありません。両接続とも 5250 アプリケーションへのアクセスを提供しますが、WebFacing サーバーへの接続では、オンライン・トランザクション処理 (OLTP) 能力 (対話式 CPW) を必要としません。このため、TN5250 セッション接続に関連した対話式機能の要件を排除することで、HATS と 5250 アプリケーションを使用するためのコスト全体を大幅に削減できます。

© Copyright IBM Corp. 2003, 2015

WebFacing サーバーへの接続を使用可能にするには、「新規プロジェクト」ウィザード、「新規接続」ウィザード、接続エディターのいずれかで、「**タイプ**」フィールドから「**5250W**」接続タイプを選択します。WebFacing サーバー接続のデフォルトのポートは、**4004** です。詳細については、158 ページの『基本』を参照してください。

既存のプロジェクトを使用可能にして WebFacing 接続オプションを活用するには、接続エディターで「**5250W**」を選択します。WebFacing サーバー接続は、プロジェクトのデフォルト接続またはバックグラウンド接続として機能を果たすことができます。

WebFacing サーバーに接続するには、5250 ホスト・システム上でサーバー・プロセスを実行することが必要です。WebFacing サーバーを始動するには、オペレーティング・システムのコマンド行で `STRTCPSVR SERVER(*WEBFACING)` というコマンドを入力します。

WebFacing サーバーを停止するには、オペレーティング・システムのコマンド行で `ENDTCPSVR SERVER(*WEBFACING)` というコマンドを入力します。

WebFacing サーバーが実行されていることを確認するには、コマンド行で `WRKACTJOB JOB(QQF*)` というコマンドを入力して、すべての WebFacing ジョブをリストで表示します。

WebFacing サーバー・ジョブである QQFWFSVR と、WebFacing 仮想端末サーバー・ジョブである QQFVTSVR が両方とも表示されます。ジョブのリストには、通常、複数の QQFVTSVR ジョブが表示されます。

注: システム値 QAUTOVRT を確認してください。QAUTOVRT の値により、自動的に開始できる VT ジョブの数が決まります。WebFacing サーバーは、VT ジョブに依存しています。この値が 0 の場合、ブラウザー・セッションは起動できません。必要な場合は、QAUTOVRT の値を *NOMAX または 0 より大きいいずれかの値に設定してください。QAUTOVRT の値を変更するには、WRKSYSVAL システム・コマンドを使用してシステム値を操作します。WebFacing サーバー接続ジョブは、QINTER サブシステムの下にもリストが表示され、INT (対話式) というマークが付いていますが、これらはサーバーの対話能力の一部とはみなされません。

制限

HATS とホスト・アプリケーション間の 5250W 接続については、Secure Socket Layer (SSL) はサポートされていません。

5250W 接続では、ワークステーション ID の事前の命名はサポートされていません。接続前に具体的なワークステーション ID を指定することはできません。

WebFacing アプリケーションとの HATS 相互運用性

概要

WebFacing アプリケーションとの HATS 相互運用性により、データ記述仕様 (DDS) ディスプレイ・ファイル・ソース・メンバー変換や、 5250 データ・ストリーム変換を、同じ Web アプリケーションの内部で実行できる機能が提供されま

す。この機能が特に有益なのは、HATS ツールを使用して変換したホスト・アプリケーションと対話する必要がある DDS ソース・ベースの変換アプリケーションを、WebFacing ツールを使用して作成したユーザーです。この機能は、DDS ディスプレイ・ファイル・ソース・メンバーに基づいて、WebFacing ツールによって提供される変換を利用する HATS ユーザーにも役立ちます。

この文脈では、相互運用性とは、1 つの HATS Web アプリケーションが、1 つの WebFacing アプリケーションと相互に運用されるか、または連携して動作することを意味します。結合 Web アプリケーションは、何らかの制限がある状態で 2 つのツールの機能を備えていますが、Web アプリケーションの内部では、これらの機能が明確に分離されています。結合 Web アプリケーションを作成するには、1 つの HATS Web プロジェクトを 1 つの WebFacing プロジェクトにリンクして、1 つのエンタープライズ・アプリケーション、つまり .ear ファイルにパッケージ化します。このエンタープライズ・アプリケーションには、両製品のランタイムの他に、5250 バックエンド・ホストへの共用接続を作成して対話する場合に使用する共通ランタイム・コンポーネントが内蔵されています。

HATS アプリケーションと WebFacing アプリケーションの相互運用を実施する方法について詳しくは、IBM Host Access Transformation Services Education Assistant の Web サイト (http://publib.boulder.ibm.com/infocenter/ieduasst/rtnv1r0/topic/com.ibm.iea.hats/plugin_coverpage.html) にある「*Create a Linked WebFacing and HATS Application*」チュートリアルを参照してください。この機能の詳細情報およびデモンストレーションについては、「IBM Enterprise Modernization Demos」(http://rational.dfw.ibm.com/atdemo/atdemo_hats_wf_recorded.html) を参照してください。

注: この機能を使用するには、HATS WebFacing フィーチャーがインストールされている必要があります。

第 21 章 セキュリティーおよび Web 高速ログオン

HATS はアプリケーションにセキュリティーを提供します。HATS は、SSL (Secure Sockets Layer) を使用して、HATS アプリケーションと接続先 Telnet サーバー間のセキュア接続を提供します。

HATS は、VT 接続用のセキュア・シェル (SSH) もサポートします。

Web 高速ログオン (WEL) は、ユーザーを認証する (HATS Web アプリケーションおよびポートレットを実行する) 機能を備えており、ユーザーは WEL によってシングル・サインオンが可能になります。

Kerberos サービス・チケットを使用すると、5250 Telnet セッションを使用し、Windows ドメイン・クライアントで稼働するリッチ・クライアント・アプリケーションのサインオンを自動化できます。

HATS は、Java 2 セキュリティーをサポートすることにより、WebSphere Application Server システム内部のシステム・リソースや API の保護機能を提供するのに役立ちます。

HATS 管理コンソールを使用する場合の WebSphere Application Server のセキュリティーの効果については、443 ページの『第 18 章 HATS 管理コンソールの使用』を参照してください。

WebSphere Application Server セキュリティー情報については、http://www.ibm.com/software/webservers/appserv/was/library/ の WebSphere Application Server オンライン・ライブラリーを参照してください。

SSL セキュリティーの使用可能化

Web アプリケーションの場合、ユーザーのブラウザーと HATS アプリケーション間で SSL セキュリティーを確保するには、HTTPS 接続が必要です。その場合、HTTP サーバーと WebSphere Application Server で HTTPS がサポートされるように構成する必要があります。HTTP サーバーの証明書は、ブラウザーの証明書ストアに保管され、ブラウザーと HTTP サーバーとの接続に使用されます。

このセクションで後述する HATS SSL 構成は、HATS アプリケーションと (SSL ポートを使用して構成する必要がある) Telnet サーバー間で SSL を構成するときに使用します。 この HATS SSL 構成のサポート対象は、HATS Web、リッチ・クライアント、EJB アプリケーション、および HATS ポートレットです。

注: HATS と WebFacing サーバー間の 5250W 接続については、SSL はサポートされていません。

HATS アプリケーションと Telnet サーバー間の SSL を使用可能にするには、接続エディターの「**セキュリティー**」タブで、「**SSL を使用可能にする**」チェック・ボックスを選択します。詳しくは、166 ページの『セキュリティー』を参照してくだ

© Copyright IBM Corp. 2003, 2015

さい。接続に対して SSL を使用可能にすることにより、接続を経由して流れるデータを暗号化し、接続の機密保護を要求します。

「**JSSE を使用**」チェック・ボックスを選択すると、HATS と HOST システムの間の接続で、SSLite の代わりに、Java Secure Socket Extension (JSSE) セキュリティー・ライブラリーを使用した TLS v1.0、TLS v1.1、または TLS v1.2 が使用可能になります。選択しない場合は (デフォルト・オプション)、SSLite ライブラリーが使用され、この接続に TLS v1.1 と TLS v1.2 は使用できません。

注: IETF Internet-Draft「TLS ベースの Telnet セキュリティー」は、TLS ベースの Telnet 接続を介して SSL ハンドシェークを行うためのプロトコルを定義しています。接続先の Telnet サーバーがこのプロトコルをサポートしている場合、接続定義の拡張接続設定に SSLTelnetNegotiated プロパティーを追加する必要があります。拡張接続設定は、接続エディターの「**拡張**」タブにあります (161 ページのオプションの拡張接続設定を構成してくださいを参照)。プロパティーの値を **true** に設定します。

HATS では、Host On-Demand テクノロジーを採用することにより、HATS アプリケーションから 3270 および 5250 アプリケーションまでの接続を Telnet プロトコルを使用してサポートします。HATS では、Host On-Demand テクノロジーによって得られる SSL サポートを使用することにより、これらの接続のセキュリティーを確保します。 SSL 上でセキュア接続を使用すると、接続を介して流れるデータが暗号化され、したがって第三者による判読を防止できます。

保護された接続の場合、HATS アプリケーションとそれが接続されている Telnet サーバーの両方が SSL をサポートしている必要があります。接続を保護するには、Telnet サーバーがデータの暗号化に使用する証明書を提供する必要があります。

接続の確立が試行されると、HATS は Telnet サーバーから証明書を受け取り、接続を受諾するか拒否するかを判断します。 HATS は、内蔵している鍵ストア・ファイルを対象に、Telnet サーバーの個人証明書に一致する署名者証明書を検索します。HATS 鍵ストア・ファイルには、Verisign、Thawte、RSA などの一連の既知の証明書が含まれています。Telnet サーバーで有効な既知の証明書を使用している場合、その証明書は、HATS で提供される既知の証明書のいずれかと一致するので、受け入れられます。この場合には、証明書を格納している鍵ストア・ファイルを作成する必要はありません。必要な署名者の証明書は、既に HATS 内蔵の鍵ストア・ファイル内に存在するためです。

Telnet サーバーが既知の証明書を使用していない場合は、有効な署名者の証明書を格納している鍵ストア・ファイルを作成して、HATS に対して構成する必要があります。 この証明書を入手するには、Telnet サーバーの鍵ストア・ファイルを開き、証明書をバイナリー .der ファイルとして抽出し、証明書管理ツール (別名 IBM 鍵管理ツール) を使用して、HATS で使用する鍵ストア・ファイルに .der ファイルをインポートします。

例えば、Telnet サーバー・プラットフォームが IBM 鍵管理ツールをサポートしている場合、Telnet サーバーの鍵ストア・ファイルから証明書ファイルを抽出するには、以下の手順を実行します。

1. IBM 鍵管理ツールを開始します。

2. 「**鍵データベース・ファイル (Key Database File)**」をクリックして、「**開く**」を選択します。IBM CMS 鍵ストア・ファイルを開く方法については、 504 ページの『IBM CMS 鍵ストア・ファイルの使用』を参照してください。
3. Telnet サーバーの鍵ストア・ファイルに対応する「**鍵データベース・タイプ**」を選択し、「**参照**」をクリックして、このファイルを格納しているディレクトリーを探します。
4. 「**OK**」をクリックします。
5. 「**鍵データベースの内容**」の下で、ドロップダウン・リストから「**署名者証明書**」を選択します。
6. 抽出する証明書を選択して、「**抽出**」をクリックします。
7. 「**データ型**」で、「**Binary DER データ (Binary DER data)**」を選択します。証明書が ASCII フォーマットの場合、「**BASE64 エンコード ASCII データ**」を選択します。
8. 証明書ファイルの名前と場所を指定して、「**OK**」をクリックします。

Telnet サーバーの鍵ストア・ファイルから抽出した証明書ファイルを収容した鍵ストア・ファイルを作成して、HATS と組み合わせて使用するには、以下の手順を実行します。

1. Telnet サーバーの鍵ストア・ファイルから抽出した証明書を HATS 開発システムにコピーします。
2. 「**スタート**」>「**すべてのプログラム**」> 「*IBM Rational SDP パッケージ・グループ*」 >「**IBM Rational HATS V9.5**」>「**Certificate Management**」をクリックします (*IBM Rational SDP パッケージ・グループ* はインストールした Rational SDP パッケージ・グループの名前です)。
3. 「**鍵データベース・ファイル**」をクリックして、「**新規**」を選択する。
4. 「**鍵データベース・タイプ (Key database type)**」には、「**PKCS12**」または「**JKS**」 (JSSE を使用している場合) を選択します。ファイルに .p12 拡張子または .jks 拡張子 (JSSE を使用している場合) の付いた名前を指定し、ファイルの場所を指定して、「**OK**」をクリックします。
5. パスワードを入力して確認し、「**OK**」をクリックします。JKS ファイルのパスワードは、少なくとも 6 文字にする必要があります。
6. 「**鍵データベース・コンテンツ**」の下で、ドロップダウン・リストから「**署名者証明書**」を選択し、「**追加**」をクリックします。
7. 「**データ型**」で、「**Binary DER データ (Binary DER data)**」を選択します。証明書が ASCII フォーマットの場合、「**BASE64 エンコード ASCII データ**」を選択します。
8. 「**参照**」をクリックして、Telnet サーバーの鍵ストア・ファイルから抽出した証明書を検索して選択し、「**OK**」をクリックします。
9. 証明書のラベルを入力して「**OK**」をクリックします。
10. 証明書管理ツールを終了します。

注: 証明書管理について詳しくは、 498 ページの『HATS アプリケーションに対する IBM Certificate Management の使用』を参照してください。

HATS が証明書を必要とする場合は、接続エディターの「**セキュリティー**」タブのオプションを使用すると、証明書を格納している鍵ストア・ファイルの HATS による検索方法を構成できます。鍵ストア・ファイルをプロジェクトにインポートする場合は、「**インポート**」ボタンを使用します。 または、鍵ストア・ファイルをプロジェクト内には格納しないが、ターゲット・ランタイム・システム上の別の場所に存在するようにすることを指定する場合は、「**特定のパスで PKCS12 鍵ストアを使用**」オプションを使用します。このオプションは、HATS アプリケーションを再展開する必要なく鍵ストア・ファイルを更新する場合に便利です。鍵ストア・ファイルをインポートすると、このファイルは、HATS によって EAR プロジェクトのルート (Web プロジェクトの場合)、¥Web Content¥WEB-INF ディレクトリー (ポートレット・プロジェクトの場合)、またはランタイム拡張機能プラグイン (リッチ・クライアント・プロジェクトの場合) のいずれかにコピーされます。インポートが完了すると、鍵ストア・ファイルは HATS プロジェクトの一部となり、プロジェクトをエクスポートする場合には、残りのプロジェクト・ファイルとともにパッケージ化されます。詳しくは、166 ページの『セキュリティー』を参照してください。

注:

1. 複数の証明書を 1 つの鍵ストア・ファイルに追加できます。
2. 複数の HATS プロジェクトで同じ鍵ストア・ファイルを使用できます。このためには、同じ鍵ストア・ファイルを EAR プロジェクト (Web アプリケーションの場合) またはランタイム拡張機能プラグイン (リッチ・クライアント・アプリケーションの場合) にインポートするか、「**特定のパスで PKCS12 鍵ストアを使用**」オプションで同じ鍵ストア・ファイルを参照します。
3. 単一の HATS プロジェクト内部の各接続が参照する鍵ストア・ファイルは、同じファイルでも別のファイルでも構いません。

WebSphere Application Server のセキュリティーについて詳しくは、http://www.ibm.com/software/webservers/appserv/was/library/ を参照してください。IBM HTTP Server のセキュリティーについては、http://www.ibm.com/software/webservers/httpservers/library.html を参照してください。

SSH セキュリティーの使用可能化

VT 接続用のセキュア・シェル (SSH) を構成するには、次のようにします。

1. 「HATS Projects」ビューで、「接続」フォルダーを展開し、VT 接続の項目をダブルクリックします。
2. 接続エディターで、「**拡張**」タブをクリックします。
3. 「オプションの拡張接続設定を構成してください」で、「**追加**」をクリックします。
4. 「パラメーターを追加」ダイアログで、「名前」フィールドのドロップダウン・リストから、「**SecurityProtocol**」パラメーターを選択し、「値」フィールドを SESSION_PROTOCOL_SSH に設定し、「**OK**」をクリックします。
5. 「**追加**」をクリックします。
6. 「パラメーターを追加」ダイアログで、「名前」フィールドのドロップダウン・リストから、「**userID**」パラメーターを選択し、「値」フィールドをご使用のユーザー ID に設定し、「**OK**」をクリックします。
7. 「**追加**」をクリックします。

8. 「パラメーターを追加」ダイアログで、「名前」フィールドのドロップダウン・リストから、「**userPassword**」パラメーターを選択し、「値」フィールドをご使用のパスワードに設定し、「**OK**」をクリックします。

Web 高速ログオン (WEL) の使用

概要

Web 高速ログオン (WEL) により、ユーザーは自分のネットワーク・セキュリティー信任状のみを使用してホスト・アプリケーションにアクセスできるようになります。これにより、HATS アプリケーションは、ネットワーク・セキュリティー層によって事前に認証されたユーザーのネットワーク信任状情報を受け入れ、その情報を使用して、使用するホスト信任状を生成できるため、HATS ユーザーはホストのログオン画面にナビゲートする必要がありません。

WEL は、企業のセキュリティーを維持するための企業のネットワーク・セキュリティー・アプリケーションとともに機能し、ユーザーがユーザー ID およびパスワードを再入力せずにホスト・システムにログオンできるようにします。WEL には、以下の利点があります。

- **使いやすい**: ユーザーは、ネットワーク・セキュリティー・アプリケーションにログオンし、ID およびパスワードを再入力せずにホスト・アプリケーションにアクセスできます。
- **パスワード関連サポート呼び出しの削減**: ユーザーがパスワードを忘れたり間違えたりして企業のサポート担当者を呼び出す頻度が減少します。
- **生産性の向上**: ユーザーは、ユーザー ID、パスワード、および認証を定義する方法が複数ある環境に 1 回だけログオンできます。

証明書高速ログオンというタイプもサポートされています。これにより、ネットワーク・セキュリティー・アプリケーションを使用するのではなく X.509 証明書を使用して認証できます。

アーキテクチャー

次の図は、ホスト Credential Mapper の例として、DCAS および RACF® を使用した WEL アーキテクチャーを示しています。

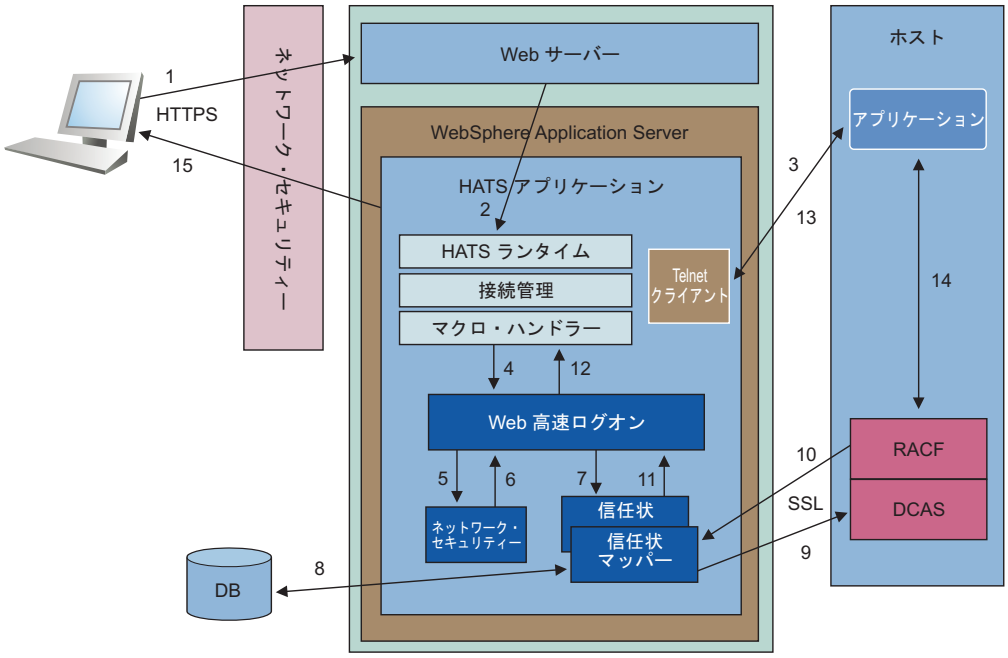


図 77. WEL アーキテクチャー

1. ユーザーはネットワーク・セキュリティー・アプリケーション (NSA) を通じて HATS アプリケーションへの接続要求を送信します。

   NSA は X.509 証明書またはユーザー ID とパスワードを使用してユーザーを認証します。NSA は認証の後、ユーザーに関連するポリシー情報に基づいて、要求されたアプリケーションへのアクセスを許可します。次に、NSA は、Web サーバー経由でユーザーのネットワーク信任状 (ネットワーク ID) を WebSphere Application Sever に渡します。

   NSA の例としては、IBM Tivoli Access Manager があります。
2. WebSphere Application Server は要求を HATS に送付します。
3. HATS ランタイムはホストへの Telnet 接続を開始します。
4. HATS マクロ・ハンドラーは WEL ログオン・マクロを実行します。 484 ページの『WEL ログオン・マクロの作成方法』を参照してください。WEL は、ホスト信任状 (ホストのユーザー ID とパスワード/パスチケット) を返すように要求する Java 関数呼び出しをそのマクロから受け取ります。
5. WEL は適切なネットワーク・セキュリティー・プラグインを呼び出し、NSA によって保存されていた場所からユーザーのネットワーク ID を取得します。

   使用されている NSA に対応するネットワーク・セキュリティー・プラグインが HATS に構成されている必要があります。 485 ページの『ネットワーク・セキュリティー・プラグイン』を参照してください。
6. ネットワーク・セキュリティー・プラグインはユーザーのネットワーク ID を返します。
7. WEL は、適切な Credential Mapper プラグインを呼び出し、ユーザーのネットワーク ID とホスト・アプリケーション ID をホスト信任状 (ホストのユーザー ID とパスワードまたはパスチケット) に変換します。

使用されている Credential Mapper に対応する Credential Mapper プラグインが HATS に構成されている必要があります。 485 ページの『Credential Mapper プラグイン』を参照してください。

8. Credential Mapper プラグインは、ユーザーのネットワーク ID をホスト・ユーザー ID にマップするための機能 (この場合は、IBM DB2® などの JDBC アクセス可能データベース) を呼び出します。

   HATS 提供の Credential Mapper プラグインは、JDBC アクセス可能データベースを使用するように設計されています。LDAP ディレクトリーを使用するという選択肢もあります。ただし、LDAP を使用する場合は、お客様独自のカスタム・プラグインを作成する必要があります。詳しくは、「*HATS Web アプリケーション・プログラマーズ・ガイド*」の章『Web 高速ログオンのプラグインの作成』を参照してください。
9. Credential Mapper プラグインは、ホストのユーザー ID とホスト・アプリケーション ID を使用してバックエンドの Credential Mapper (この場合は、DCAS および RACF) を呼び出し、パスチケットを要求します。
10. RACF がパスチケットを生成して返します。
11. Credential Mapper プラグインはホストのユーザー ID とパスチケットを WEL に返します。
12. WEL はホストのユーザー ID とパスチケットをマクロに返します。
13. マクロはホストのユーザー ID とパスチケットをホストのログオン画面に挿入します。
14. ホスト・アプリケーションはホストのユーザー ID とパスチケットを検証し、接続を許可します。
15. ユーザーに対してホスト・アプリケーション画面が提示されます。

インプリメンテーションの計画

HATS プロジェクトで WEL を使用するためのセットアップおよび構成を計画するときに考慮する必要がある事項があります。以下に、必要な情報のリストを示します。

- ホスト・タイプ
- ユーザーが経由するネットワーク・セキュリティー層
- WEL を使用してユーザーがアクセスするホスト・アプリケーション
- それらのアプリケーションに必要なホスト認証
- ホスト認証信任状の格納先およびアクセス方法
- 組み込み Credential Mapper プラグインがジョブの実行に十分であるかどうか、またはカスタム・プラグインを作成する必要があるかどうか

インプリメンテーション

WEL のセットアップを開始するための情報を入手したら、以下のステップを実行する必要があります。

- ネットワーク・セキュリティー・アプリケーションを (使用する場合は) 構成します。
- WEL 用に HATS プロジェクトの接続を構成します。

「**接続**」エディターの「**セキュリティー**」タブにある「**Web 高速ログオンを使用する**」チェック・ボックスを選択し、「**構成 ...**」ボタンをクリックすると、Web 高速ログオン・エディターが開きます。このエディターには以下の 3 つのタブがあります。

  - **ネットワーク・セキュリティー・プラグイン**
  - **Credential Mapper プラグイン**
  - **ソース**

- 以下のようにして WEL を構成します。
  - 使用するネットワーク・セキュリティー・プラグインを識別し、初期化パラメーターを指定します。
  - EAR プロジェクトで使用する Credential Mapper プラグイン、各プラグインの選択基準、および各プラグインの初期化パラメーターを識別します。選択基準に基づいて最適な順序に配列します。
- バックエンド信任状ソース (DCAS など) を作成して設定します。
- DCAS などのバックエンド信任状ソースに接続するために必要な必須鍵ストア・ファイルを構成します。証明書管理ツール (IBM 鍵管理ツールとも呼ばれる) が提供されています。詳しくは、494 ページの『SSL 鍵ストア・ファイルの作成 (DCAS のみ)』を参照してください。
- WEL 処理を起動するためのユーザー ID およびパスワードを要求するプロンプトを含むマクロを記録します。
- HATS プロジェクト (接続のログオン・マクロなど) で必要なときにこれらのマクロを起動します。

WEL ログオン・マクロの作成方法

WEL ログオン・マクロを作成するには、以下のステップを行います。

1. 「**HATS Projects**」ビューで、「**接続**」フォルダーからプロジェクト接続を選択します。
2. HATS ツールバーで、「**HATS ホスト端末をオープン**」アイコンをクリックして、セッションを開始します。
3. 「**マクロの記録**」アイコンをクリックします。
4. ユーザー ID 入力フィールドを含む画面にナビゲートします。
5. ツールバーから「**プロンプト・アクションを追加**」アイコンを選択します。「**プロンプト・アクションを追加**」ウィザードが表示されます。フィールドに入力します。詳しくは、387 ページの『プロンプト・アクションを追加』を参照してください。
6. 「**プロンプト・アクションを追加**」ダイアログで「**Web 高速ログオンを使用**」を選択します。「**ユーザー ID**」のプロンプト・タイプを選択し、「アプリケーション ID」フィールドにアプリケーション ID を入力します。
7. 「**パスワード**」入力フィールドにナビゲートします。
8. 「**プロンプト・アクションを追加**」アイコンを選択し、「**Web 高速ログオンを使用**」と「**パスワード**」プロンプト・タイプを選択して、アプリケーション ID を入力します。
9. ログイン処理が完了したら、「**マクロを停止**」アイコンおよび「**マクロを保存**」をクリックします。

マクロを記録して保存したら、このマクロを起動するように HATS を構成する必要があります。いくつかの方法を使用できます。

- WEL ログオン・マクロを接続の接続マクロとして定義します。WEL を接続マクロとして構成するには、以下のステップを行います。
  1. 「**HATS Projects**」ビューから接続を選択します。
  2. 「**マクロ**」タブをオープンします。
  3. 「**接続マクロ**」ドロップダウン・リストを使用して WEL マクロを選択します。
- 接続イベントの「**マクロを実行**」オプションで WEL ログオン・マクロを起動します。
  1. 「プロジェクト設定」ビューで「**イベント**」タブを選択します。
  2. 「アプリケーション・イベント」の下にある「**接続**」をクリックします。
  3. 接続イベントの「概要」パネルで「**アクション**」タブをクリックします。
  4. 「アクション」パネルで「**追加**」をクリックします。
  5. 「アクションを選択」ページで「**マクロを実行**」を選択して「**次へ**」をクリックします。
  6. 「アクション・プロパティーの定義」ページのドロップダウン・リストから WEL ログオン・マクロを選択して「**完了**」をクリックします。
- 画面カスタマイズの「**マクロを実行**」オプションで WEL ログオン・マクロを起動します。詳しくは、192 ページの『マクロを実行』を参照してください。
- マクロから統合オブジェクトを作成します。

ネットワーク・セキュリティー・プラグイン

WEL には以下のネットワーク・セキュリティー・プラグインを選択できます。 .ear ファイルごとに指定可能なセキュリティー・プラグインは 1 つのみです。「**プラグイン・タイプ**」の横にあるドロップダウン・リストから選択し、以下のリストを表示させます。

- なし (証明書高速ログオンと同様に、ネットワーク・セキュリティー・パッケージを使用しない場合に使用します)
- カスタム (「*HATS Web アプリケーション・プログラマーズ・ガイド*の章『Web 高速ログオンのプラグインの作成』を参照)
- アクセス・マネージャー・ネットワーク・セキュリティー
- WebSphere Portal ネットワーク・セキュリティー・プラグイン

  注: WebSphere Portal ネットワーク・セキュリティー・プラグインは、HATS ポートレット・プロジェクトを操作している場合にのみ表示されます。

Credential Mapper プラグイン

WEL には以下の組み込み Credential Mapper プラグインを選択できます。「**追加**」ボタンをクリックし、「**組み込み Credential Mapper プラグインの追加**」を選択した後、以下のリストから選択します。

- DCAS/RACF/JDBC Credential Mapper

注: パスチケットの取得に、DCAS および RACF が zOS オペレーティング・システムとともに使用されます。ユーザーのネットワーク ID をユーザーのホスト ID にマップするために、JDBC アクセス可能リポジトリーが必要です。

- 証明書ベースの DCAS/RACF Credential Mapper

  注: パスチケットの取得に、DCAS および RACF が zOS オペレーティング・システムとともに使用されます。このプラグインの場合、証明書は直接 DCAS に渡され、ホスト ID とパスチケットのペアが返されるため、JDBC アクセス可能リポジトリーは必要ありません。

- JDBC ボールト Credential Mapper

  注: DB2 や Oracle、さらには Windows 上の Excel スプレッドシートなど、任意の JDBC/ODBC 準拠のリポジトリーを使用できます。このリポジトリーは、ホストのユーザー ID とパスワードを保管するために使用されます。

- WebSphere Portal Credential Vault Credential Mapper

  注:

  1. このプラグインは、ポートレット・プロジェクトの「**組み込み資格情報マッパー・プラグインの追加**」ダイアログにのみ表示されます。
  2. このマッパーは、指定された信任状ボールト・スロットからユーザー ID とパスワードのペアを取得します。
  3. この Credential Mapper プラグインには、**SLOT_ID** (デフォルトは **HATS**) および **SLOT_TYPE** (デフォルトは **2**) の 2 つの設定が必要です。
  4. スロット ID は、信任状ボールトで使用されるスロット名の接頭部を指定します。完全なスロット名は次のように構成されます。
     - スロット ID + (スペース) + ホストの完全名 + (スペース) + アプリケーション ID
     - 例: `HATS zserveros.demos.ibm.com CICSA`
  5. アプリケーション ID が存在しない場合、スロットにアクセスする際に、ホスト名の後ろに末尾スペースが必要です。
  6. スロット・タイプは、信任状ボールト・スロットのタイプを指定します。
     - 0 プライベート
     - 1 共用
     - 2 管理 (デフォルト)
     - 3 システム
  7. ポータル管理者は、スロット ID およびスロット・タイプをセットアップする責任があります。
  8. 詳しくは、http://www.ibm.com/developerworks/websphere/zones/portal/proddoc/index.html において、ご使用の WebSphere Portal レベルの資料を参照してください。

- テスト Credential Mapper

  注: このプラグインは WEL マクロのテスト用に提供されています。これは、ユーザーから提供されたハードコーディングのホスト・ユーザー ID とパスワードを使用するため、HATS Toolkit 内でのテスト専用です。

「**カスタム Credential Mapper プラグインの追加**」 を選択してカスタム Credential Mapper プラグインの追加を選択し、テキスト・ボックスに完全修飾プラグイン名を入力する方法もあります。カスタム・プラグインの作成について詳しくは、「*HATS Web アプリケーション・プログラマーズ・ガイド*」の章『Web 高速ログオンのプラグインの作成』を参照してください。

Credential Mapper プラグインを選択すると、クラス、名前、説明、作成者などの詳細情報が「**詳細**」セクションに自動的に入力されます。「**初期化**」セクションには、選択したプラグイン用に構成されている一連のパラメーターが表示されます。「**追加**」ボタンをクリックして、使用するプラグインの追加パラメーターを指定できます。また、「**除去**」を選択すると、選択したパラメーターを除去できます。不要なパラメーターのみを除去できます。

Credential Mapper の選択

表 9 では、どの Credential Mapper 要求がこのプラグインによって処理されるかを説明しています。

表 9. 認証タイプと説明

認証タイプ	説明
AuthType_All	すべての認証タイプで使用する信任状を識別します
AuthType_3270	3270 エミュレーションで使用する信任状を識別します
AuthType_5250	5250 エミュレーションで使用する信任状を識別します
AuthType_VTHost	VT エミュレーションで使用する信任状を識別します

複数の Credential Mapper が定義されている場合、2 次選択基準としてホスト・マスクを使用して、使用する最適な Credential Mapper を識別できます。表 10 は、ホスト・マスクの指定方法の例を示しています。

表 10. 一致するホスト・マスクと値

ホスト・マスク	一致する値
*.raleigh.ibm.com	.raleigh.ibm.com で終了するすべてのアドレスに一致します
ralvm*	ralvm で開始するすべてのアドレスに一致します
*	すべてに一致します
xyz	xyz を含むすべてのホスト・アドレスに一致します

初期化パラメーター

z/OS および DCAS を使用するソリューションの場合は、DCAS プラグイン・パラメーターを追加します。これらのパラメーターを追加すると、CMP は、ネットワーク ID をホスト ID にマップして、ホストで実行される DCAS アプリケーションからパスチケットを取得できます。 パスチケットはパスワードに似た信任状ですが、

パスチケットは特定の時間が経過すると有効期限が切れてしまい、1 回しか使用できません。DCAS には、IBM Resource Access Control Facility (RACF) セキュリティー・サーバーなどのパスチケット生成をサポートする SAF (Security Access Facility) 準拠のサーバー製品が必要です。

注: DCAS サーバーへのセキュア接続を確保するため、HATS DCAS/RACF/JDBC および証明書ベースの DCAS/RACF Credential Mapper プラグインにより、Java Secure Socket Extension (JSSE) が使用されます。 このサポートの HATS V7 より前のバージョンからのマイグレーションについては、25 ページの『Web 高速ログオン (WEL)』を参照してください。

DCAS/RACF/JDBC Credential Mapper プラグインの DCAS パラメーター

必須の DCAS パラメーター: Credential Mapper プラグインで DCAS サーバーに安全に接続するには、以下のパラメーターをいくつか組み合わせることが必要です。

CMPI_DCAS_TRUSTSTORE
このパラメーターには、HATS DCAS クライアント証明書と DCAS サーバー証明書をルックアップするために使用する鍵ストア・ファイルの名前が記述されています。CMPI_DCAS_USE_DEFAULT_TRUSTSTORE が **true** に設定されている場合は、このパラメーターによって指定される鍵ストア・ファイルの代わりに、JSSE のデフォルトの鍵ストア・ファイルが使用されます。

CMPI_DCAS_TRUSTSTORE_PASSWORD
このパラメーターには、CMPI_DCAS_TRUSTSTORE で指定されている鍵ストア・ファイルのパスワードが記述されています。

CMPI_DCAS_TRUSTSTORE_TYPE
このパラメーターには、CMPI_DCAS_TRUSTSTORE で指定されている鍵ストア・ファイルのタイプが記述されています。有効値は **pkcs12**、**jceks**、および **jks** です。

CMPI_DCAS_USE_DEFAULT_TRUSTSTORE
このパラメーターは、HATS DCAS クライアント証明書と DCAS サーバー証明書をルックアップするために、JSSE のデフォルトの鍵ストア・ファイルを使用するかどうかを示します。デフォルトは **false** です。**true** に指定すると、CMPI_DCAS_TRUSTSTORE パラメーターによって指定される鍵ストア・ファイルの代わりにこの鍵ストア・ファイルが使用されます。

注: JSSE デフォルトの鍵ストア・ファイルの検索順序は、以下のとおりです。

- javax.net.ssl.trustStore システム・プロパティーで指定した場所
- <java-home>/lib/security/jssecacerts
- <java-home>/lib/security/cacerts

以下のパラメーターは、JDBC データベース Credential Mapper とともに機能するように設計されています。このタイプのネットワーク・アクセス可能なデータベースを使用すると、ユーザーのネットワーク ID をホスト ID に柔軟かつ安全に関連付けることができます。関連したアクセス情報をすべて格納することによって、既存

のデータベースへのアクセスを構成することも、新規作成されたデータベースを指すこともできます。データベースのセキュリティーのレベルは、データベース・ベンダーによって異なります。

CMPI_DCAS_DB_ADDRESS
これは、データベースのアドレスを提供する URL ストリングです。

CMPI_DCAS_DB_NET_DRIVER
このストリングは、ネットワーク・データベース・ドライバーとして機能するクラスの名前を含みます。例えば、このストリングは COM.ibm.db2.jdbc.net.DB2Driver となります。このクラスの場所は既存のクラス・パスにあると想定されます。

CMPI_DCAS_DB_USERID
これは、データベースにアクセスするときに使用するユーザー・アカウントの ID です。

CMPI_DCAS_DB_CASE_SENSITIVE
このパラメーターでは、DCAS プラグインでユーザーのアプリケーション ID とネットワーク ID を小文字に変換し、lcase() メソッドを使用して HCM データベースに SQL 照会を行うかどうかを指定します。lcase() メソッドをサポートしない SQL アプリケーションを使用する場合、このパラメーターを true に設定する必要があります。

CMPI_DCAS_DB_PASSWORD
これは、データベースにアクセスするときに使用するユーザー・アカウントのパスワードです。

CMPI_DCAS_DB_TABLE
このエントリーは、必要な照会に使用するテーブルを識別します。

以下の 4 つのパラメーター値は、Credential Mapper データベースの列名に一致し、列の内容を明確に示さなければなりません。IBM DB2 などの一部のデータベースでは、データベースの 4 つの列見出しはすべて大文字でなければなりません (例: NETWORKID、HOSTADDRESS、APPLICATIONID、HOSTID)。

CMPI_DCAS_DB_NETID_COL_NAME
このエントリーは、ネットワーク ID 値 (NETWORKID) を含む列の名前を識別します。

CMPI_DCAS_DB_HOSTADDR_COL_NAME
このエントリーは、ホスト・アドレス値 (HOSTADDRESS) を含む列の名前を識別します。

CMPI_DCAS_DB_HOSTAPP_COL_NAME
このエントリーは、ホスト・アプリケーション値 (APPLICATIONID) を含む列の名前を識別します。

注: アプリケーション ID は 3270 ホスト・タイプの場合にのみ使用されます。

CMPI_DCAS_DB_HOSTID_COL_NAME
このエントリーは、ユーザーのホスト識別値 (HOSTID) を含む列の名前を識別します。

上記のパラメーターによって指定された情報に基づいて、データベースの SQL 照会を作成してホスト ID を取得できます。この照会は、ネットワーク ID、ホスト・アドレス、およびホスト・アプリケーションを照会のキーとして使用します。結果はホスト識別の列で示されます。照会が正常に終了すると、DCAS サーバーの呼び出しが実行され、パスチケットが要求されます。

オプションの DCAS パラメーター: 以下の DCAS パラメーターはオプションです。

CMPI_DCAS_DB_PRESERVE_WHITESPACE
このパラメーターは、信任状要求パラメーターから空白文字を除去するかどうかを示します。true の場合は、空白文字を除去しません。デフォルトは false です。

CMPI_DCAS_HOST_ADDRESS
デフォルト DCAS ホスト・アドレスは、HATS 接続に指定された宛先ホストに基づいて判別されます。

CMPI_DCAS_HOST_PORT
DCAS ホスト・アドレスは、要求で指定された宛先ホストに基づいて判別されます。デフォルトのポート・アドレスとしては 8990 が使用されますが、このパラメーターを使用してオーバーライドできます。

CMPI_DCAS_NO_FIPS
このパラメーターを true に設定すると、FIPS セキュリティー・プロバイダーを使用しないことが指定されます。代わりに、デフォルトのセキュリティー・プロバイダーが使用されます。デフォルトは **false** です。

CMPI_DCAS_REQUEST_TIMEOUT
このパラメーターは、パスチケット要求タイムアウトをミリ秒単位で指定します。これは、マクロ・タイムアウト値よりも小さくしてください。デフォルトは **50000** です。

CMPI_DCAS_TRACE_LEVEL
このパラメーターは、DCAS プラグインのトレース・レベルを指定します。トレース・メッセージは HATS トレース・ファイルに記録されます。トレース・レベル値には、以下の設定があります。

- **0 = なし**: トレースを実行しません。これはデフォルトです。
- **1 = 最小**: API、パラメーター、戻り値、およびエラーをトレースします。
- **2 = 通常**: 最小のトレースに加え、内部 API、パラメーター、および通知メッセージもトレースします。
- **3 = 最大**: 通常のトレースに加え、Java 例外もトレースします。

CMPI_DCAS_USE_NETID_AS_HOSTID
true に設定すると、ホスト ID を対象とするデータベース検索はスキップされます。このパラメーターは、ネットワーク ID が RACF ID である場合にも使用します。デフォルトは **false** です。

CMPI_DCAS_VERIFY_SERVER_NAME
このパラメーターは、証明書の検証に加え、証明書のサーバー・ホスト名を検証するかどうかを示します。デフォルトは **false** です。

証明書ベースの DCAS/RACF Credential Mapper プラグインの DCAS パラメーター

必須の DCAS パラメーター: Credential Mapper プラグインで DCAS サーバーに安全に接続するには、以下のパラメーターをいくつか組み合わせることが必要です。

CMPI_DCAS_TRUSTSTORE
このパラメーターには、HATS DCAS クライアント証明書と DCAS サーバー証明書をルックアップするために使用する鍵ストア・ファイルの名前が記述されています。CMPI_DCAS_USE_DEFAULT_TRUSTSTORE が **true** に設定されている場合は、このパラメーターによって指定される鍵ストア・ファイルの代わりに、JSSE のデフォルトの鍵ストア・ファイルが使用されます。

CMPI_DCAS_TRUSTSTORE_PASSWORD
このパラメーターには、CMPI_DCAS_TRUSTSTORE で指定されている鍵ストア・ファイルのパスワードが記述されています。

CMPI_DCAS_TRUSTSTORE_TYPE
このパラメーターには、CMPI_DCAS_TRUSTSTORE で指定されている鍵ストア・ファイルのタイプが記述されています。有効値は **pkcs12**、**jceks**、および **jks** です。

CMPI_DCAS_USE_DEFAULT_TRUSTSTORE
このパラメーターは、HATS DCAS クライアント証明書と DCAS サーバー証明書をルックアップするために、JSSE のデフォルトの鍵ストア・ファイルを使用するかどうかを示します。デフォルトは **false** です。**true** に指定すると、CMPI_DCAS_TRUSTSTORE パラメーターによって指定される鍵ストア・ファイルの代わりにこの鍵ストア・ファイルが使用されます。

注: JSSE デフォルトの鍵ストア・ファイルの検索順序は、以下のとおりです。

- javax.net.ssl.trustStore システム・プロパティーで指定した場所
- <java-home>/lib/security/jssecacerts
- <java-home>/lib/security/cacerts

オプションの DCAS パラメーター: 以下の DCAS パラメーターはオプションです。

CMPI_DCAS_HOST_ADDRESS
デフォルトの DCAS ホスト・アドレスは、HATS 接続に指定された宛先ホストです。

CMPI_DCAS_HOST_PORT
デフォルトのポート・アドレスとしては 8990 が使用されますが、このパラメーターを使用してオーバーライドできます。

CMPI_DCAS_NO_FIPS
このパラメーターを true に設定すると、FIPS セキュリティー・プロバイダーを使用しないことが指定されます。代わりに、デフォルトのセキュリティー・プロバイダーが使用されます。デフォルトは **false** です。

CMPI_DCAS_REQUEST_TIMEOUT
このパラメーターは、パスチケット要求タイムアウトをミリ秒単位で指定します。これは、マクロ・タイムアウト値よりも小さくしてください。デフォルトは 50000 です。

CMPI_DCAS_TRACE_LEVEL
このパラメーターは、DCAS プラグインのトレース・レベルを指定します。トレース・メッセージは HATS トレース・ファイルに記録されます。トレース・レベル値には、以下の設定があります。

- **0 = なし**: トレースを実行しません。これはデフォルトです。
- **1 = 最小**: API、パラメーター、戻り値、およびエラーをトレースします。
- **2 = 通常**: 最小のトレースに加え、内部 API、パラメーター、および通知メッセージもトレースします。
- **3 = 最大**: 通常のトレースに加え、Java 例外もトレースします。

CMPI_DCAS_VERIFY_SERVER_NAME
このパラメーターは、証明書の検証に加え、証明書のサーバー・ホスト名を検証するかどうかを示します。デフォルトは false です。

JDBC ボールト Credential Mapper プラグインのボールト・パラメーター

CMPIVaultPlugin に対するボールト・パラメーターを追加します。JDBC ベースのボールト・ホスト・セキュリティーを使用する環境の場合は、ボールト・プラグイン・パラメーターを追加します。このモデルは、DCAS パスチケット環境でネットワーク ID およびホスト ID の関連付けに使用するデータベース・メカニズムと同一です。唯一の違いは、ボールト・スタイル認証では CMPI_VAULT_DB_HOSTPW パラメーターが必要な点です。

必須ボールト・パラメーター: 以下のボールト・パラメーターは必須です。

CMPI_VAULT_DB_ADDRESS
これは、データベースのアドレスを提供する URL ストリングです。例えば、このストリングは `jdbc:db2://dtagw:6789/CMTEST` となります。

CMPI_VAULT_DB_NET_DRIVER
このストリングは、ネットワーク・データベース・ドライバーとして機能するクラスの名前を含みます。例えば、このストリングは `COM.ibm.db2.jdbc.net.DB2Driver` となります。このクラスの場所は既存のクラス・パスにあると想定されます。

CMPI_VAULT_DB_USERID
これは、データベースにアクセスするときに使用するユーザー・アカウントの ID です。

CMPI_VAULT_DB_CASE_SENSITIVE
このパラメーターでは、ボールト・プラグインでユーザーのアプリケーション ID とネットワーク ID を小文字に変換し、lcase() メソッドを使用して HCM データベースに SQL 照会を行うかどうかを指定します。lcase() メソッドをサポートしない SQL アプリケーションを使用する場合、このパラメーターを true に設定する必要があります。

CMPI_VAULT_DB_PASSWORD
これは、データベースにアクセスするときに使用するユーザー・アカウントのパスワードです。

CMPI_VAULT_DB_TABLE
このエントリーは、必要な照会に使用するテーブルを識別します。

以下の 5 つのパラメーター値は、Credential Mapper データベースの列名に正確に一致していなければなりません。

CMPI_VAULT_DB_NETID_COL_NAME
このエントリーは、ネットワーク ID 値 (NETWORKID) を含む列の名前を識別します。

CMPI_VAULT_DB_HOSTADDR_COL_NAME
このエントリーは、ホスト・アドレス値 (HOSTADDRESS) を含む列の名前を識別します。

CMPI_VAULT_DB_HOSTAPP_COL_NAME
このエントリーは、ホスト・アプリケーション値 (APPLICATIONID) を含む列の名前を識別します。

注: アプリケーション ID は 3270 ホスト・タイプの場合にのみ使用されます。

CMPI_VAULT_DB_HOSTID_COL_NAME
このエントリーは、ユーザーのホスト識別値 (HOSTID) を含む列の名前を識別します。

CMPI_VAULT_DB_HOSTPW_COL_NAME
このエントリーは、ユーザーのホスト・パスワード (PASSWORD) を含む列の名前を識別します。

上記のパラメーターによって指定された情報に基づいて、データベースの SQL 照会を作成してホスト ID を取得できます。この照会は、ネットワーク ID、ホスト・アドレス、およびホスト・アプリケーションを照会のキーとして使用します。結果はホスト識別の列で示されます。照会が正常に終了すると、ユーザー ID およびパスワードが戻されます。

オプションのボールト・パラメーター: 以下のボールト・パラメーターはオプションです。

CMPI_VAULT_TRACE_LEVEL
このパラメーターは、ボールト・プラグインのトレース・レベルを指定します。トレース・メッセージは HATS トレース・ファイルに記録されます。トレース・レベル値には、以下の設定があります。

- **0 = なし**: トレースを実行しません。これはデフォルトです。
- **1 = 最小**: API、パラメーター、戻り値、およびエラーをトレースします。
- **2 = 通常**: 最小のトレースに加え、内部 API、パラメーター、および通知メッセージもトレースします。
- **3 = 最大**: 通常のトレースに加え、Java 例外もトレースします。

CMPI_VAULT_DB_PRESERVE_WHITESPACE
このパラメーターは、信任状要求パラメーターから空白文字を除去するかどうかを示します。true の場合は、空白文字を除去しません。デフォルトは false です。

WebSphere Portal 信任状ボールトの Credential Mapper のボールト・パラメーター

注: この初期化パラメーターは、HATS ポートレット・プロジェクトを処理している場合だけに表示されます。

SLOT_ID
これは、ボールト・スロットから受動ユーザー・パスワード信任状を取得するボールト・スロット ID です。

SLOT_TYPE
このパラメーターは、アクセスする信任状スロットのタイプを示します。有効値は、以下のとおりです。

- **0 = プライベート・スロット**
- **1 = 共用スロット**
- **2 = 管理スロット** (デフォルト値)
- **3 = システム・スロット**

各タイプのスロットについては、WebSphere Portal 製品情報 Web サイト (http://www.ibm.com/developerworks/websphere/zones/portal/proddoc.html) を参照してください。

SSL 鍵ストア・ファイルの作成 (DCAS のみ)

DCAS サーバーと通信するには、クライアント認証を使用して SSL 接続を確立する必要があります。このためには、鍵ストア・ファイルを指定する必要があります。サポートされている鍵ストア・ファイル・タイプは、PKCS12、JKS、または JCEKS です (PKCS12 は、Solaris ではサポートされていません)。
CMPI_DCAS_TRUSTSTORE パラメーターで指定する鍵ストア・ファイルを作成するには、証明書管理ツール (IBM 鍵管理ツールとも呼ばれる) を使用します。この鍵ストア・ファイルには、HATS DCAS クライアントの証明書と DCAS サーバーの証明書 (公開鍵) の情報が記述されている必要があります。

注:

1. CMPI_DCAS_USE_DEFAULT_TRUSTSTORE パラメーターを **true** に設定すると、CMPI_DCAS_TRUSTSTORE パラメーターによって指定される鍵ストア・ファイルの代わりに JSSE のデフォルトの鍵ストア・ファイルが使用されます。デフォルトのファイルには、HATS DCAS クライアントの証明書と DCAS サーバーの証明書 (公開鍵) の情報が記述されている必要があります。
2. HATS DCAS クライアントの証明書は、SSL クライアント認証用に DCAS サーバーの鍵ストア・ファイルに追加/インポートする必要もあります。

IBM Key Management ツールを使用して作成された古い証明書が既に存在している場合は、その証明書をインポートできます。旧システムで作成した個人用サーバー証明書は、旧システムからエクスポートすることも、新システムにインポートすることもできません。ただし、これを実行するための方法があります。

1. 既存の .kdb ファイルを新規の鍵ストア・ファイル (PKCS12、JKS、または JCEKS) にインポートします。
2. 証明書 (DCAS 個人用サーバー証明書など) を .p12 形式の証明書にエクスポートします。
3. 証明書 (.p12 形式) を新規の鍵ストア・ファイル (PKCS12、JKS、または JCEKS) にインポートします。

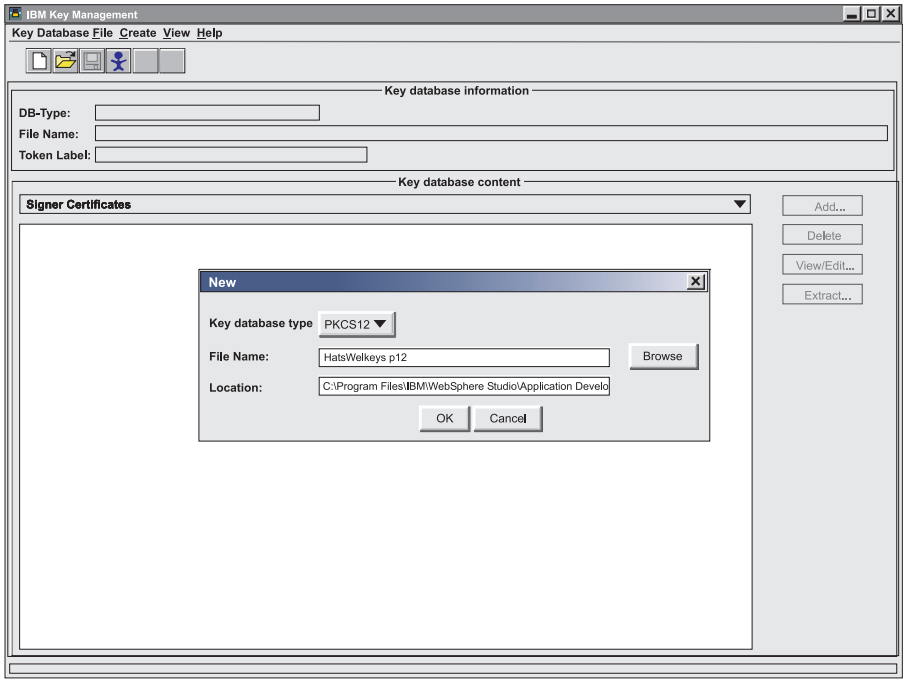


図 78. HATS 証明書管理

CMPI_DCAS_TRUSTSTORE パラメーターで指定される HatsWelkeys.p12 という名前の新規の鍵ストア・ファイルを作成するには、以下の手順を実行します。

注: 以下の手順では、PKCS12 鍵ストア・ファイルの作成方法を示します。 HATS アプリケーションのターゲット・プラットフォームが Solaris である場合は、以下の **PKCS12** ではなく、**JCEKS** または **JKS** のいずれかの鍵データベース・タイプを使用してください。

1. 「**スタート**」>「**すべてのプログラム**」> 「*IBM Rational SDP パッケージ・グループ*」 >「**IBM Rational HATS V9.5**」>「**Certificate Management**」をクリックします (*IBM Rational SDP パッケージ・グループ* はインストールした Rational SDP パッケージ・グループの名前です)。
2. 「**鍵データベース・ファイル**」をクリックして、「**新規**」を選択する。鍵データベースのタイプには、「**PKCS12**」を選択します。「ファイル名」および「場所」に入力するか、デフォルト値を確定します。
3. 「**OK**」をクリックします。
4. パスワードを入力して確認し、「**OK**」をクリックします。

5. DCAS サーバーの証明書を鍵ストア・ファイルに追加します。「**鍵データベースの内容**」に、「**署名者証明書**」が表示されていることを確認してください。 それ以外の場合は、プルダウン・メニューを選択して変更します。その後、「**追加**」を選択します。
   a. 「**データ・タイプ (Data type)**」には「**Binary DER データ (Binary DER data)**」を選択する。サーバー証明書が ASCII 形式の場合は「**base64 エンコードの ASCII データ (Base64-encoded ASCII data)**」を選択します。
   b. 「**証明書ファイル名 (Certificate file name)**」 フィールドにファイル名を入力します。
   c. 「**場所**」フィールドにパス名を入力します。
   d. 「**OK**」をクリックします。
   e. 証明書のラベルを入力して「**OK**」をクリックします。
6. DCAS クライアントの証明書を鍵ストア・ファイルに追加します。
   a. 「**鍵データベースの内容 (Key database content)**」を「**個人証明書 (Personal Certificates)**」に変更し、「**エクスポート/インポート (Export/Import)**」をクリックします。
   b. 「鍵のエクスポート/インポート (Export/Import Key)」パネルで、「**鍵をインポート (Import Key)**」を選択します。
   c. 鍵ファイル・タイプに「**PKCS12**」を選択します。
   d. 「**ファイル名**」フィールドにクライアント証明書の .p12 ファイル名を入力し、「**場所**」フィールドにパス名を入力します。

      注: インポートする証明書を含む鍵ストア・ファイル (.p12/pkcs12) を参照し、ユーザー ID とパスワードを入力してファイルを開きます。鍵ストア・ファイルには、インポートする証明書だけを格納するようにします。.kdb ファイルから証明書をインポートすることもできます。その場合、各証明書ごとに選択できます。
   e. 「**OK**」をクリックし、パスワードを入力してソース鍵データベースを開きます。
   f. 「**OK**」をクリックします。
7. 証明書管理 GUI を終了します。

注: 証明書管理ツールの詳細については、498 ページの『HATS アプリケーションに対する IBM Certificate Management の使用』を参照してください。

Kerberos サービス・チケットの使用

Kerberos サービス・チケットを使用して、5250 Telnet セッションを使用し、Windows ドメイン・クライアントで稼働するリッチ・クライアント・アプリケーションのサインオンを自動化できます。そのためには、ご使用のリッチ・クライアント・アプリケーションの 5250 Telnet 接続の接続ファイル (.hco ファイル) を編集します。「**セキュリティー**」タブをクリックして、 **Kerberos サービス・チケットを使用してサインオンを自動化する (Windows ドメイン・クライアントのみ)** ためのオプションを選択します。

このサポートは、IBM i Kerberos ベース・ネットワーク認証とともに、基礎となる IBM Host On-Demand 接続ベースの自動化を使用します。詳しくは、Host On-Demand Knowledge Center において、接続ベースの自動化を構成する方法に関するセクション (http://publib.boulder.ibm.com/infocenter/hodhelp/v11r0/index.jsp?topic=/com.ibm.hod.doc/doc/logon/logon14.html) を参照してください。

Java 2 セキュリティー

Java 2 セキュリティーは、ファイル入出力、ソケット、環境プロパティーなど、特定の保護システム・リソースにアプリケーションがアクセスできるようにする前に、アクセス権を検査するポリシー・ベースのアクセス制御機構を提供します。

Java 2 セキュリティーは、アプリケーション・サーバーによってサポートされます。Java 2 セキュリティーは、デフォルトでは使用不可になっていますが、グローバル・セキュリティーを使用可能にするように WebSphere セキュリティーを構成すると自動的に使用可能になります。Java 2 セキュリティーは、WebSphere グローバル・セキュリティーを使用可能にすると使用可能になりますが、使用不可にすることもできます。Java 2 セキュリティーとグローバル・セキュリティーは、互いに独立して構成できます。グローバル・セキュリティーを使用不可にしても、Java 2 セキュリティーが自動的に使用不可になるわけではありません。Java 2 セキュリティーは、明示的に使用不可にする必要があります。

HATS が Java 2 セキュリティーをサポートするのは、サポートされている任意のサーバー・ソフトウェア・プラットフォーム上で WebSphere Application Server が実行されている場合です。HATS でサポートされるサーバー・ソフトウェアについて詳しくは、「Rational Host Access Transformation Services: System Requirements」(http://www.ibm.com/support/docview.wss?uid=swg27011794) を参照してください。

HATS は、以下の環境では Java 2 セキュリティーをサポートしません。

- HATS Toolkit でのサーバーで実行 (ROS) 機能。
- Windows 以外のプラットフォームでの WebSphere Portal

ポリシー・ファイル

HATS Toolkit には、Java 2 セキュリティーが使用可能になったときに動作するように、展開されている HATS アプリケーションに対して必要なアクセス権を付与する Java 2 セキュリティー・ポリシー・ファイルが組み込まれています。HATS の基本的な動作は、Java 2 セキュリティーを使用可能にした状態でテストされていますが、一部の HATS アプリケーションでは、Java 2 セキュリティーを使用可能にした状態で動作できるようにするために、ポリシー・ファイルの変更が必要な場合があります。 例えば、アプリケーション内部のビジネス・ロジックがシステム・リソースにアクセスする場合や、鍵リング・ファイルなどのファイルが、まだポリシー・ファイルの対象範囲になっていないパスに存在する場合などです。

HATS アプリケーションのポリシー・ファイルは、HATS パースペクティブの「**ナビゲーター**」ビューで参照できます。HATS アプリケーションの EAR プロジェクトを展開して、META-INF フォルダーを開き、was.policy ファイルを探します。HATS では、作成した新規プロジェクトごとに、デフォルトのポリシー・ファイル

がコピーされます。デフォルトのポリシー・ファイルを変更する場合、このファイルは、インストールされている HATS プラグインの下位で、以下のディレクトリーに置かれています。

```
<shared_install_directory>¥plugins¥com.ibm.hats_nnn¥predefined¥
projects¥earProject¥META-INF¥
```

ここで、*shared_install_directory* は IBM Installation Manager を使用して HATS オファリングをインストールした共用リソース・ディレクトリー、*nnn* は HATS のバージョンおよびビルド・レベルです。

Java 2 セキュリティーおよびポリシー・ファイルの詳細については、WebSphere Knowledge Center (http://publib.boulder.ibm.com/infocenter/wasinfo/v8r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/tsec_enablejava2sec.html) を参照してください。

HATS アプリケーションに対する IBM Certificate Management の使用

IBM Key Management ツールに付属の Certificate Management を使用すれば、Secure Sockets Layer (SSL) 接続に必要な証明書を作成および管理できます。

HATS の場合、SSL は以下の状況で使用されます。

- HATS アプリケーションと Telnet サーバーの間で、セキュア接続を確保するため。
- Digital Certificate Access Server (DCAS) とクライアントの間で、ホスト信任状を取得するために HATS アプリケーションで Web 高速ログオン (WEL) を使用するため。

鍵データベース・ファイルの作成

HATS アプリケーション用に SSL を構成する前に、鍵ストア・ファイルまたはトラストストア・ファイルとも呼ばれる、鍵データベース・ファイルを作成する必要があります。

新規の鍵データベース・ファイルを作成するには、以下の手順を実行します。

1. 「**スタート**」>「**すべてのプログラム**」> 「*IBM Rational SDP パッケージ・グループ*」 >「**IBM Rational HATS V9.5**」>「**Certificate Management**」をクリックします (*IBM Rational SDP パッケージ・グループ* は、インストールした Rational SDP パッケージ・グループの名前です)。
2. これにより、IBM Key Management ツールが起動します。
3. 「**鍵データベース・ファイル (Key Database File)**」>「**新規**」をクリックします。
4. 鍵データベース・タイプとして「**PKCS12**」、「**JKS**」、または「**JCEKS**」鍵データベース・ファイルを選択します。

   注: HATS アプリケーションと Telnet サーバーの間のセキュア接続には、PKCS12 が必要です。
5. ファイル名を入力します。HATS は、特定のファイル名を必要としません。
6. 場所についてディレクトリー名を入力します。HATS は、鍵データベース・ファイルを作成するために特定のディレクトリーを必要としませんが、これを使用す

るアプリケーションを展開するには、鍵データベース・ファイルをエンタープライズ・アーカイブ・ファイル (.ear ファイル) にコピーする必要があります。

7. 「**OK**」をクリックします。
8. パスワードを入力して確認し、「**OK**」をクリックします。

既存の鍵データベース・ファイルを開くには、以下の手順を実行します。

1. 「**鍵データベース・ファイル (Key Database File)**」>「**開く**」をクリックします。
2. 鍵データベース・タイプを選択します。
3. 「**参照**」をクリックして、鍵データベース・ファイルを参照します。
4. 鍵データベース・ファイルを選択して、「**開く**」をクリックします。
5. 「**OK**」をクリックします。
6. 現行パスワードを入力して「**OK**」をクリックします。

鍵データベース・ファイルを作成または開いた後、以下の操作が可能です。

- 定義済みの既知の認証局 (CA) からの証明書を要求する。この手順の場合は、サーバー証明書の発行元である CA の識別に必要な CA 署名者証明書で鍵データベース・ファイルが事前構成されているため、それほど多くの構成作業は必要ありません。 『認証局からの証明書の要求および保管』を参照してください。
- 未知の認証局からの証明書を要求する。この手順では、CA の署名者証明書も取得して鍵データベース・ファイルに保管する必要があるため、より多くの構成が必要です。さらに、ホスト信任状の取得元となる SSL パートナーが署名者証明書を使用できるようにしておく必要があります。『認証局からの証明書の要求および保管』を参照してください。
- 自己署名証明書を作成する。この手順の場合、認証局は必要ありません。サーバーのインストール後即時に使用できます。ただし、SSL パートナーが自己署名証明書を使用できるようにしておく必要があります。認証局から証明書が取得されるまでの間のテスト用に、この手順を使用することもできます。 502 ページの『自己署名証明書の使用』を参照してください。
- SSL パートナーと証明書を交換する。構成によっては、SSL 通信を使用可能にするために、Telnet または DCAS サーバーなど、任意の SSL パートナーがサーバーの鍵データベース・ファイル内の証明書を使用できるようにしておく必要があります。 502 ページの『証明書の交換』を参照してください。

注: 実行中の HATS アプリケーションで使用されている鍵データベース・ファイルを変更する場合は、必ず HATS アプリケーションを停止して再始動してください。

認証局からの証明書の要求および保管

鍵データベース・ファイルを作成するときに、サーバー証明書の発行元である CA の識別に必要となる既知の CA の CA 署名者 (トラステッド・ルート) 証明書で鍵データベース・ファイルが事前構成されます。鍵データベース・ファイルに署名者証明書が事前定義されていない CA は、未知の CA と呼ばれます。新しく作成された鍵データベース・ファイルには、以下の既知の CA 署名者証明書が自動的に保管され、トラステッド・ルート証明書としてマークが付けられます。

- Thawte Personal Premium CA

- Thawte Personal Freemail CA
- Thawte Personal Basic CA
- Thawte Premium Server CA
- Thawte Server CA
- RSA secure server CA (VeriSign からも取得)
- VeriSign class 4 public primary CA
- VeriSign class 3 public primary CA
- VeriSign class 2 public primary CA

以下の各セクションでは、既知の CA および未知の CA からの証明書の要求および保管のための手順について、概要を説明します。

- 証明書要求の作成
- 証明書要求の送信
- 証明書の保管

証明書要求の作成

公開/秘密鍵および証明書要求を作成するには、以下の手順を実行します。

1. IBM Key Management ウィンドウの「**鍵データベース・コンテンツ (Key database content)**」の下にあるドロップダウン・リストから「**個人証明書要求 (Personal Certificate Requests)**」を選択します。
2. 「**新規**」をクリックします。
3. データベース内の鍵および証明書の識別用の鍵ラベルを入力します。
4. 使用する鍵サイズに対応する数値を選択します。大きな鍵サイズを選択するとセキュリティーは強化されますが、接続を確立するためにはクライアントおよびサーバー上で多くの処理が必要になります。
5. 証明書を所有するサーバーの共通名 (TCP/IP ホスト名) を入力します (例: `myserver.mycompany.com`)。
6. 組織名を入力します。
7. オプションとして、組織単位、およびその他のロケーション情報を入力します。
8. 国別コードを選択します。
9. 証明書要求ファイル名を入力するか、デフォルトのファイル名を使用します。

「**OK**」をクリックすると、情報が処理され、証明書要求が作成されたディレクトリー内で、複数のファイルが作成または更新されます。鍵データベースをバックアップする場合は、これらのファイルもバックアップしてください。鍵データベース・ファイルまたは証明書要求が破損する可能性があるため、これらのファイルを編集しないでください。

証明書要求の送信

証明書要求を送信する場合は、CA の手順に従います。

以下のリストは、2 つの既知の CA の URL を示しています。

- Thawte: http://www.thawte.com/
- Verisign: http://www.verisign.com/

選択した CA 次第で、証明書管理から生成された証明書要求を E メールで送信するか、CA から提供されたフォームまたはファイルに証明書要求を組み込むことができます。

CA による証明書要求の処理を待つ間に、 502 ページの『自己署名証明書の使用』に記載された手順を使用して自己署名証明書を作成および保管して、テスト用の SSL セキュリティーを使用可能にできます。

証明書の保管

未知の CA から申請済み証明書を受け取った場合、CA に接触して CA の署名者 (ルート) 証明書を取得します。申請済み証明書を保管する前に、未知の CA の署名者証明書を鍵データベース・ファイルに保管する必要があります。CA の署名者証明書は、申請済み証明書の検証に使用されます。

未知の CA の署名者証明書を保管するには、未知の CA の署名者証明書のバックアップ・コピーを作成した後に、次の手順を実行します。

1. IBM Key Management ウィンドウの「**鍵データベース・コンテンツ (Key database content)**」の下にあるドロップダウン・リストから「**署名者証明書 (Signer Certificates)**」を選択します。
2. 「**追加**」をクリックします。
3. データ・タイプ「**BASE64 エンコード ASCII データ (BASE64 encoded ASCII data)**」(armored 64 形式) を選択します。
4. 証明書ファイル名を入力します。
5. 証明書の場所またはパスを入力します。
6. 「**OK**」をクリックします。ファイルは、トラステッドとしてマークが付けられて保管されます。

既知の CA または未知の CA から受け取った申請済み証明書を保管するには、証明書のバックアップ・コピーを作成した後、以下の手順を実行します。

1. ドロップダウン・リストから「**個人証明書 (Personal Certificates)**」を選択し、「**受信 (Receive)**」をクリックして、証明書要求を受け取ります。「ファイルから証明書を受信 (Receive Certificate from a File)」ウィンドウが表示されます。
2. データ・タイプは「**BASE64 エンコード ASCII データ (BASE64-encoded ASCII data)**」(armored 64 形式) でなければなりません。
3. 証明書ファイル名を入力します。
4. 証明書の場所 (パス名) を入力します。
5. 「**OK**」をクリックします。保管したばかりの証明書が最初の項目として表示されます。
6. 鍵情報を表示する場合は、証明書を強調表示して「**表示/編集 (View/Edit)**」をクリックします。
7. 証明書の名前が「個人証明書 (Personal Certificate)」ドロップダウン・リストの下に表示され、「個人証明書要求 (Personal Certificate Requests)」ドロップダウン・リストからその証明書要求が消去されます。
8. デプロイできるように、鍵データベース・ファイルをエンタープライズ・アーカイブ・ファイル (.ear ファイル) にコピーします。エンタープライズ・アーカイブが実行中の場合は、停止して再始動します。

自己署名証明書の使用

自己署名証明書を作成および使用するには、以下の手順を使用します。

1. IBM Key Management ウィンドウの「**鍵データベース・コンテンツ (Key database content)**」の下にあるドロップダウン・リストから「**個人証明書 (Personal Certificates)**」を選択します。
2. 「**新規の自己署名 (New Self-signed)**」をクリックします。
3. データベース内の鍵および証明書の識別用の鍵ラベルを入力します。
4. 証明書バージョンとして「**X509 V3**」を選択します。
5. 使用する鍵サイズに対応する数値を選択します。大きな鍵サイズを選択するとセキュリティーは強化されますが、接続を確立するためにはクライアントおよびサーバー上で多くの処理が必要になります。
6. 証明書を所有するサーバーの共通名 (TCP/IP ホスト名) を入力します (例: `myserver.mycompany.com`)。
7. 組織名を入力します。
8. オプションとして、組織単位、およびその他のロケーション情報を入力します。
9. 国別コードを選択します。
10. 自己署名証明書を有効にする日数を入力します。
11. 「**OK**」をクリックします。
12. デプロイできるように、鍵データベース・ファイルをエンタープライズ・アーカイブ・ファイル (.ear ファイル) にコピーします。エンタープライズ・アーカイブが実行中の場合は、停止して再始動します。

証明書の交換

構成によっては、DCAS サーバーなどの SSL パートナーも証明書を使用できるようにしておく必要があります。ご使用のサーバーで未知の CA からの証明書を使用している場合、未知の CA の署名者 (ルート) 証明書を、SSL パートナーが使用できるようにしておく必要があります。ご使用のサーバーで自己署名証明書を使用している場合、自己署名証明書のコピーを、SSL パートナーが使用できるようにしておく必要があります。

交換するために証明書ファイルを作成するには、以下の手順を実行します。

1. 鍵データベース・ファイルを開きます。 498 ページの『鍵データベース・ファイルの作成』を参照してください。
2. 証明書を抽出します。
   - ご使用のサーバーが不明 CA から発行された証明書を使用している場合は、以下のようにします。
     a. 「**鍵データベースの内容**」の下で、ドロップダウン・リストから「**署名者証明書**」を選択します。
     b. ご使用のサーバー用の証明書を発行した CA の署名者 (ルート) 証明書を強調表示します。
     c. 「**抽出**」をクリックします。

- ご使用のサーバーが自己署名証明書を使用している場合は、以下のようにします。
  a. 「**鍵データベースの内容 (Key database content)**」の下で、ドロップダウン・リストから「**個人証明書 (Personal Certificate)**」を選択します。
  b. ご使用のサーバーで使用している証明書を強調表示します。
  c. 「**証明書の抽出 (Extract Certificate)**」をクリックします。

3. 「証明書をファイルに抽出 (Extract Certificate to a File)」ウィンドウで、「**Base64 エンコード ASCII データ (Base64-encoded ASCII data)**」または「**バイナリー DER データ (Binary DER data)**」を選択します。一般的に、証明書が E メールで安全に転送される場合は、「**Base64 エンコード ASCII データ (Base64-encoded ASCII data)**」を使用します。証明書のファイル名および場所は自由に選択できます。
4. 「**OK**」をクリックして、証明書ファイルを抽出します。
5. 証明書ファイルを SSL パートナー (例えば DCAS サーバー) に安全に転送し、証明書をその鍵データベース・ファイルに追加します。

証明書管理ツールの考慮事項

DOS ウィンドウ

証明書管理ツールを起動すると、DOS ウィンドウが開き、ツールが実行されている間は表示され続けます。このウィンドウは、ツールを終了すると閉じます。DOS ウィンドウが表示されるのは、証明書管理ツールのデフォルトの実行プロパティーが「**通常のウィンドウ**」であるためです。

ツールの起動時に DOS ウィンドウを最小化するには、以下のステップを実行します。

1. 「**スタート**」>「**すべてのプログラム**」> 「*IBM Rational SDP パッケージ・グループ*」 >「**IBM Rational HATS V9.5**」にアクセスします (*IBM Rational SDP パッケージ・グループ* は、インストールした Rational SDP パッケージ・グループの名前です)。
2. 「**証明書の管理**」を右クリックし、「**プロパティー**」を選択する。
3. 実行プロパティーが「**最小化**」になるように構成する。
4. 「**OK**」をクリックします。

管理者として実行

Windows Vista では、「**管理者として実行**」オプションを使用してプログラムを開始しようとすると、証明書管理ツールは開始されません。これは、以下の 2 つの方法で発生します。

1. 「証明書管理」アイコンを右クリックし、「**管理者として実行**」を選択する。
2. 「証明書管理」アイコンのプロパティーの拡張設定で「**管理者として実行**」を指定して、プログラムを実行するアイコンをダブルクリックする。

証明書管理は、「**管理者として実行**」オプションを使用せずに実行できます。

IBM CMS 鍵ストア・ファイルの使用

IBM CMS 鍵ストア・ファイルを使用するには、適切なプロバイダー・クラスをプロバイダー・リストに追加して登録する必要があります。これを行うには、次の手順を実行します。

1. 次の場所にあるプロバイダー・リストを編集します。

   <*RationalSDP_install_directory*>¥jdk¥jre¥lib¥security¥java.security

   RationalSDP_install_directory は、IBM Installation Manager を使用して Rational SDP オファリングをインストールしたディレクトリーです。

2. 次の記述をプロバイダーのリストに追加します。

   security.provider.<*n*>=com.ibm.security.cmskeystore.CMSProvider

   <*n*> は、優先順位です。例えば、以下のようになります。

   security.provider.10=com.ibm.security.cmskeystore.CMSProvider

第 22 章 言語サポート

HATS のユーザー・インターフェースおよびコンテキストに依存したヘルプは、以下の言語で提供されています。

- アラビア語
- ブラジル・ポルトガル語
- 中国語 (簡体字)
- 中国語 (繁体字)
- チェコ語
- 英語
- フランス語
- ドイツ語
- ハンガリー語
- イタリア語
- 日本語
- 韓国語
- ポーランド語
- ロシア語
- スペイン語
- トルコ語

すべての言語が単一の製品イメージとしてインストールされます。各国語サポートはオペレーティング・システムに依存するため、使用する言語用の適切なフォントとキーボード・サポートをオペレーティング・システムにインストールしておく必要があります。例えば、フランス語のフォントとキーボード・サポートをまだインストールしていないのに、ホスト・セッション言語としてフランス語を使用しようとしても、文字を正しく表示できない場合があります。

注:

1. **HATS Toolkit** の「スタート」メニュー項目は、HATS をインストールした Rational SDP パッケージ・グループの下の IBM Rational HATS V9.5 グループにあります。
2. **HATS Toolkit** の「スタート」メニュー項目を使用する場合、HATS Toolkit は、Rational SDP 用にインストールされた言語で翻訳されるものとして、その同じ言語で開始されます。
3. アラビア語を使用した HATS の開始について詳しくは、534 ページの『HATS Toolkit の画面方向』を参照してください。

© Copyright IBM Corp. 2003, 2015

言語コード

言語コードは、ユーザーに提示する言語を決定するために製品の随所で使用されます。言語および言語コードについては、表 11 を参照してください。

表 11. HATS 言語コード

言語	言語コード
アラビア語	ar
チェコ語	cs
ドイツ語	de
英語	en
スペイン語	es
フランス語	fr
ハンガリー語	hu
イタリア語	it
日本語	ja
韓国語	ko
ポーランド語	pl
ブラジル・ポルトガル語	pt_BR
ロシア語	ru
トルコ語	tr
中国語 (簡体字)	zh
中国語 (繁体字)	zh_TW

コード・ページ

HATS は以下のコード・ページをサポートしています。各 HATS プロジェクト用のコード・ページは、プロジェクトの作成時に選択することができます。また、後でプロジェクト・エディターを使用して変更することもできます。

表 12. コード・ページ

コード・ページ	場所または用途
037	ベルギー ブラジル カナダ オランダ ポルトガル 米国
273	オーストリア ドイツ
274	ベルギー (旧)
275	ブラジル (旧)
277	デンマーク ノルウェー

表 12. コード・ページ (続き)

コード・ページ	場所または用途
278	フィンランド スウェーデン
280	イタリア
284	スペイン ラテンアメリカ (スペイン語)
285	英国
297	フランス
420	アラビア語
424	ヘブライ語 (新規コード)
500	マルチリンガル
803	ヘブライ語 (旧コード)
838	タイ語
870	ボスニア・ヘルツェゴビナ クロアチア チェコ ハンガリー ポーランド ルーマニア スロバキア スロベニア
871	アイスランド
875	ギリシャ
924	マルチリンガル ISO ユーロ
930	日本語 (カタカナ) 日本 (拡張カタカナ)
933	韓国
937	台湾 (中国語 (繁体字)) 注: アクセント付き文字のサポートについては、 509 ページの『コード・ページ 937 用のアクセント付き文字の使用』 を参照してください。
939	日本語 (拡張ローマ字)
1025	ベラルーシ ブルガリア マケドニア共和国 ロシア セルビア/モンテネグロ (キリル文字)
1026	トルコ
1047	オープン版
1112	ラトビア リトアニア
1122	エストニア
1123	ウクライナ
1137	ヒンディ語

表 12. コード・ページ (続き)

コード・ページ	場所または用途
1140	ベルギー・ユーロ ブラジル・ユーロ カナダ・ユーロ オランダ・ユーロ ポルトガル・ユーロ 米国ユーロ
1141	オーストリア・ユーロ ドイツ・ユーロ
1142	デンマーク・ユーロ ノルウェー・ユーロ
1143	フィンランド・ユーロ スウェーデン・ユーロ
1144	イタリア・ユーロ
1145	ラテンアメリカ・ユーロ (スペイン語) スペイン・ユーロ
1146	英国ユーロ
1147	フランス・ユーロ
1148	マルチリンガル・ユーロ
1149	アイスランド・ユーロ
1153	ボスニア・ヘルツェゴビナ・ユーロ クロアチア・ユーロ チェコ・ユーロ ハンガリー・ユーロ ポーランド・ユーロ ルーマニア・ユーロ スロバキア・ユーロ スロベニア・ユーロ
1154	ベラルーシ・ユーロ ブルガリア・ユーロ マケドニア共和国ユーロ ロシア・ユーロ セルビア/モンテネグロ (キリル文字) ユーロ
1155	トルコ・ユーロ
1156	ラトビア・ユーロ リトアニア・ユーロ
1157	エストニア・ユーロ
1158	ウクライナ・ユーロ
1160	タイ・ユーロ
1166	カザフスタン・ユーロ
1364	韓国ユーロ
1371	台湾 (中国語 (繁体字)) ユーロ
1388	PRC (中国語簡体字拡張 : GB18030)
1390	日本語 (カタカナ Unicode 拡張)
1399	日本語 (Latin Unicode 拡張)

エンコードの設定

HATS JSP (HATS 変換およびテンプレートなど) で使用されている文字セットは、ユーザーのブラウザーで構成されているエンコード、およびアプリケーション・サーバーで指定されているエンコードと一致する必要があります。HATS アプリケーションは、すべてのロケールで UTF-8 文字セットを使用して、JSP を作成します。

WebSphere Application Server には、さまざまなロケール用にさまざまなデフォルト・エンコード設定があります。一部の 2 バイト文字および双方向言語データ入力では、そのロケールのエンコード設定が UTF-8 でない場合に、正しくデコードされない場合があります。HATS アプリケーションが実行されているアプリケーション・サーバーを、ご使用の HATS アプリケーションの JSP が使用するエンコードと同じエンコードを使用するよう構成する必要があります。UTF-8 を構成するには、次の手順に従います。

1. WebSphere Application Server 管理コンソールを開きます。Rational SDP に含まれるローカル・テスト環境を使用している場合は、「サーバー」ビューで、サーバーを右クリックし、「**管理**」>「**管理コンソールの実行**」を選択します。
2. 管理コンソールにログインし、左側のナビゲーション・パネルの「**サーバー**」ツリーを開いて、「**アプリケーション・サーバー**」を選択します。
3. 変更するサーバー名を選択します。
4. 右側のパネルで「**Java およびプロセス管理**」ツリーを開いて、「**プロセス定義**」を選択し、次に開いたパネルで「**Java 仮想マシン**」を選択します。
5. 「**汎用 JVM 引数**」セクションで、`-Dclient.encoding.override=UTF-8` と入力します。
6. 「**OK**」ボタンをクリックします。
7. メッセージ・ブロックの「**保存**」リンクを選択します。
8. ログオフして、WebSphere Application Server を再始動します。

5250 Unicode サポート

ユニコード・データ・ストリームを使用可能に設定するパラメーターを使用して、5250 ホスト用の接続パラメーターを構成できます。**unicodeDataStreamEnabled** パラメーターは、接続エディターの「**拡張**」タブ上にあるパラメーターです。パラメーターを追加するには、「**オプションの拡張接続設定の構成 (Configure optional, advanced connections settings)**」に移動し、「**追加**」ボタンをクリックします。プルダウンの「**名前**」 メニューから「**unicodeDataStreamEnabled**」パラメーターを選択し、「**値**」を入力します。「**OK**」をクリックします。

コード・ページ 937 用のアクセント付き文字の使用

5250 接続に中国語 (繁体字) コード・ページ 937 を使用する場合に、アクセント付き文字を使用できるようにするには、拡張接続設定パラメーター **useAccentedCharacters** を追加し、接続エディターの「**拡張**」タブでこのパラメーターの値を **true** に設定する必要があります。Host On-Demand セッション・パラメーターの追加方法については、159 ページの『拡張』を参照してください。また、

CCSID 937 ではアクセント付き文字がサポートされていないため、このサポートを使用するにはホストの CCSID が 65535 であることを確認してください。この機能は 5250 接続でのみサポートされています。

コード・ページ 1388 (GB18030) の使用

PRC (中国語簡体字拡張: GB18030) コード・ページ 1388 を使用するときの考慮事項を以下にリストします。

- GB18030 文字をテンプレートに追加する場合、「**プレビュー**」タブと「**ソース**」タブでは正しく表示できますが、テンプレート・エディターの「デザイン」ビューでは正しく表示されない場合があります。
- 画面変換でウィジェットを操作する場合、GB18030 データは Page Designer で表示されないことがあります。 GB18030 データが Page Designer で表示されない場合は、以下の手順を実行します。
  1. 「ホスト・コンポーネントを挿入」ウィザードの「レンダリング・オプション」パネルで「**ウィジェット設定**」ボタンをクリックします。
  2. 「設定」パネルで「**プロジェクトのデフォルトを使用**」ボックスをクリアします。
  3. 「スタイル」フィールドの横にある「**CSS スタイル・プロパティー**」ボタンをクリックします。
  4. GB18030 4 バイト・サポート・フォント (**SimSun-18030** など) をフォント・ファミリー・リストに追加します。
  5. 「**OK**」および「**完了**」を適宜クリックして、ウィザードを閉じます。

コード・ページ 1390 および 1399 用のホスト・コード・マッピング

日本語のホスト・コード・ページ 1390 または 1399 を使用する場合、複数の DBCS コード・ポイント上で、IBM パーソナル・コミュニケーションズ製品と Host On-Demand 製品の間で、ホストから PC へのコード・マッピングの不整合が存在しています。Host On-Demand では、これらの不整合を解消するために、**UseHodCDRA1399** パラメーターにより Unicode の再マップが制御されます。以下の表は、**UseHodCDRA1399** パラメーターの true および false の両方の設定のマッピングの要約です。

表 13. Unicode マッピングの要約

EBCDIC コード・ポイント	Unicode コード・ポイント (UseHodCDRA1399=true)	Unicode コード・ポイント (UseHodCDRA1399=false)
0x4260	U+FF0D (Fullwidth hyphen-minus)	U+2212 (Minus sign)
0xE9F3	U+2212 (Minus sign)	U+FF0D (Fullwidth hyphen-minus)
0x43A1	U+FF5E (Fullwidth Tilde)	U+301C (Wave Dash)
0xE9F4	U+301C (Wave Dash)	U+FF5E (Fullwidth Tilde)
0x447C	U+2225 (Parallel To)	U+2016 (Double Vertical Line)
0xDFE5	U+2016 (Double Vertical Line)	U+2225 (Parallel To)
0x444A	U+2015 (Horizontal Bar)	U+2014 (EM Dash)
0xDDB7	U+2014 (EM Dash)	U+2015 (Horizontal Bar)

表 13. Unicode マッピングの要約 (続き)

EBCDIC コード・ポイント	Unicode コード・ポイント (UseHodCDRA1399=true)	Unicode コード・ポイント (UseHodCDRA1399=false)
0x426A	U+FFE4 (Fullwidth Broken Bar)	U+00A6 (Broken Bar)
0xE9F5	U+00A6 (Broken Bar)	U+FFE4 (Fullwidth Broken Bar)

HATS では、同じ目的のために、Host On-Demand の **UseHodCDRA1399** パラメーターを使用します。**UseHodCDRA1399** パラメーターの HATS のデフォルトは、**false** です。パラメーターを **true** に設定するには、HATS プロジェクトで接続を編集して、次の手順を実行します。

1. 「**詳細設定**」タブをクリックします。
2. 「**追加**」をクリックします。
3. パラメーター名のドロップダウン・リストから、「**UseHodCDRA1399**」を選択します。
4. パラメーター値を「**true**」に設定します。
5. 「**OK**」をクリックします。

JIS2004 サポート

新しい日本語国別標準コード化文字セットである JIS2004 では、3 つのタイプの新規文字が追加されます。

1. JIS2004 サポートより前では使用不可であった、通常の新規文字。
2. 単一の文字を表す 2 つのリンクされた Unicode コード・ポイントである、代理文字。2 つの Unicode コード・ポイントが分けられると、それらに意味はありません。
3. 1 つの文字として結合された 2 つの Unicode コード・ポイントである、結合文字。結合文字を形成する 2 つの個々の Unicode コード・ポイントは、別々に表示することもできます。

JIS2004 サポートの考慮事項と制限

アプリケーションで JIS2004 を使用する場合は、以下の項目を考慮してください。

1. JIS2004 は、Windows Vista SP1 でのデフォルトの日本語フォント・サポートです。Vista オペレーティング・システムで継続して JIS90 を使用する場合は、Microsoft サービス・パック KB927490 をインストールします。このサービス・パックは、Windows Update を使用してダウンロードし、追加ソフトウェアとして適用できます。
2. Windows XP で JIS2004 文字を表示するには、Microsoft パッチ KB927489 をインストールする必要があります。パッチがインストールされている場合でも、代理文字などの一部の JIS2004 文字は、IME がサポートされていないため挿入できません。
3. HATS のデフォルトのフォント設定では、フォント設定を一部変更しないと、JIS2004 文字が正しく表示されません。

   HATS Web アプリケーションの場合、各 HATS ウィジェット (フィールド・ウィジェットなど) でスタイル設定を有効にして、以下のフォントを追加します。

- MS MinCho
- MS Gothic
- Meiryo

HATS リッチ・クライアント・アプリケーションの場合、各 HATS ウィジェットのフォント設定を MS Mincho フォントに設定します。

4. Internet Explorer V7 以上を使用する場合、前の項目で説明したようにフォントを設定することにより、新しい JIS2004 文字を正しくサポートすることができます。
5. Internet Explorer V6 を使用する場合、または Windows XP で実行する HATS Toolkit でプレビュー機能を使用する場合、一部の新しい JIS2004 文字はまったく正しく表示されません。
6. 代理文字のサポート:
   a. HATS 端末で、IME を使用して JIS2004 文字を入力すると、候補ウィンドウに代理文字が黒の正方形として表示されます。ただし、文字が HATS に送信されると、HATS 端末には代理文字が正しく表示されます。
   b. Host On-Demand カスタム・テーブルの場合、JIS2004 代理文字はサポートされません。
7. 結合文字のサポート:
   a. Windows アプリケーション (メモ帳など)、および Firefox バージョン 3 は、JIS2004 結合文字を 1 つの文字として正しく処理します。ただし、実行時に HATS リッチ・クライアント・アプリケーションを実行する場合、または Internet Explorer で HATS Web アプリケーションを実行する場合、結合文字は入力フィールドで 1 つの文字ではなく 2 つの別々の文字として処理されます。カーソルが 2 つの結合文字の間に配置される可能性があります。
   b. 実行時に Internet Explorer で HATS Web アプリケーションを実行するときに、入力フィールドに結合文字を入力し、「レンダリング」タブの「DBCS」セクションで「**無保護のシフトアウト/シフトイン文字はスペースとして表示**」設定が有効である場合、結合文字は白の正方形になります。この設定を無効にするか、または Firefox ブラウザーを使用することにより、白の正方形を修正できます。
   c. HATS PDF 印刷機能は、結合文字を正しく表示できません。

3270E セッションの PDT 印刷および Print-to-File の JIS2004 サポート

JIS2004 文字を印刷するには、次のようにします。

- PDT 印刷の場合、pdfpdt ファイル (ibm5577.pdfpdt など) を指定し、パラメーター KANJI_CODE=SHIFT_JIS が指定されていることを確認します。
- Print-to-File の場合、パラメーター KANJI_CODE を使用してファイル・エンコードを指定します。
  – フォント・イメージを持つ PC ファイルとして出力をダンプして、出力を PC ファイルとして保存するには、pdfpdt ファイルで KANJI_CODE=SHIFT_JIS を指定することにより、ファイル・エンコードを Shift JIS に設定します。印刷出力はネイティブ PC ファイルとして保存され、ファイルの内容を物理プリンターに出力するために使用できます。

– 代理ペアのコード・ポイントを持つ Unicode ファイルとして出力をダンプして、出力を Unicode ファイルとして保存するには、pdfpdt ファイルで KANJI_CODE=UNICODE ファイルを指定することにより、ファイル・エンコードを Unicode に設定します。印刷出力は UCS-2 ファイルであり、そのファイルを追加のデータ処理またはアーカイブ目的で使用できます。

次の表は、JIS2004 文字を持つデータを出力ファイルに印刷する (Print-to-File) ときに、さまざまな pdfpdt ファイルで KANJI_CODE セッション・パラメーターにさまざまな値を指定した結果を示しています。

表 14. KANJI_CODE セッション・パラメーターの要約

pdfpdt ファイルのセッション・パラメーター	**ASCII テキスト・モード (basic_dbcs.pdf)**	**その他のサポートされるプリンター (esc_*.pdf、ibm*.pdf、lips*.pdf など)**
KANJI_CODE=SHIFT_JIS	代理ペアはネイティブ PC ファイルに格納できないため、適用外。	出力はネイティブ PC ファイルに保存され、JIS2004 文字と UDA 文字のフォント・イメージ (バイナリー・データ) がこのファイルに格納されます。
KANJI_CODE=UNICODE	出力は Unicode ファイルに保存されます。代理ペアと UDA 文字は UCS-2 で格納されます。	PC プリンターは Unicode データを受け入れないため、適用外。

さらに、ファイルに印刷する場合、セッション・パラメーター printerMimeType および printSaveAsExtension を適切に構成する必要もあります。

表 15. Print-to-File セッション・パラメーターの要約

設定	**pdfpdt ファイルのセッション・パラメーター**	**HATS のセッション・パラメーター** *printerMimeType*	**HATS のセッション・パラメーター** *printSaveAsExtension*	**結果**
さらに印刷するための設定	KANJI_CODE=SHIFT_JIS	application/octet-stream	.bin	バイナリー・データを使用するネイティブ PC ファイル。
ASCII テキスト・モードの設定	KANJI_CODE=UNICODE	text/plain	.txt	UCS-2 ファイル。

詳しくは、 417 ページの『プロジェクト用印刷サポートの定義』および 162 ページの『印刷』を参照してください。

コード・ページ 1390 および 1399 用の JIS2004 サポートの無効化

コード・ページ 1390 および 1399 を使用した接続に対する JIS2004 サポートを無効にするには、以下の手順を実行します。

1. 「HATS Projects」ビュー内のプロジェクトを展開します。
2. 「接続」フォルダーを展開します。
3. 接続をダブルクリックして、接続エディターを開きます。

4. 「**詳細設定**」タブをクリックします。
5. 「**オプションの拡張接続設定を構成してください**」で、「**追加**」をクリックします。
6. 「パラメーターを追加」パネルで、以下の設定を入力します。
   a. 「名前」フィールドに enableJIS2004 と入力します。
   b. 「値」フィールドに false と入力します。
   c. 「**OK**」をクリックします。

注: HATS 印刷サポートを使用して作成された特定の PDF ファイルを表示できるようにするには、JIS2004 サポートを無効にする必要があります。詳しくは、417 ページの『3270E 接続の場合』を参照してください。

キーボードおよびディスプレイの文字の再マップ

HATS の Host On-Demand カスタム・テーブルのサポートにより、ホストから PC へのコード・ページ変換テーブルをカスタマイズすることで、ユーザーのキーボードおよびディスプレイの文字を再マップすることができます。カスタム・テーブル・ファイルには、キーボードおよびディスプレイ・テーブルを任意に組み合わせて、任意の数のテーブルを含めることができます。キーボード・テーブルは、ユーザーのキーボード文字を、ホスト・アプリケーション用にカスタマイズされた入力として再マップします。ディスプレイ・テーブルは、ホストのディスプレイ文字を、ユーザー用にカスタマイズされた画面出力に再マップします。

注: この再マップは、HATS キーボード・マッピングと関係がありません。この再マップは文字の再マップを処理します (例えば、a、b、c など) 。一方、HATS キーボード・マッピングは、制御キーのマッピングを処理します (例えば、Clear、Enter、F1 など)。 HATS キーボード・マッピングの詳細については、423 ページの『第 16 章 キーボード・サポートの使用可能化』を参照してください。

サンプルのカスタム・テーブル・ファイル CustomTableExample.txt が *<RationalSDP_install_directory>*¥hats¥customtables ディレクトリーにあります。ファイル・フォーマットは Java プロパティー・ファイルのフォーマットと同様です。「#」で始まる行はコメントで、n および N として表示される変数は 16 進値です。以下の例は、サンプルのカスタム・テーブル・ファイルのフォーマットを示しています。

```
##### Start of file
# Map SBCS Local nn to SBCS EBCDIC NN in keyboard table
sbcs.keyboard.0xnn=0xNN
# Map DBCS Unicode nnnn to DBCS EBCDIC NNNN in keyboard table
dbcs.keyboard.0xnnnn=0xNNNN
...
# Map SBCS EBCDIC NN to SBCS local nn in display table
sbcs.display.0xNN=0xnn
# Map DBCS EBCDIC NNNN to DBCS Unicode nnnn in display table
dbcs.display.0xNNNN=0xnnnn
...
##### End of file
```

ご使用のプロジェクトでカスタム・テーブルを使用できるようにするには、次の手順に従います。

1. 例として上記サンプルを使用してカスタム・テーブル・ファイルを作成し、**customtables** という名前のフォルダーにコピーします。このフォルダーは、ご使用の HATS Web アプリケーションの .ear プロジェクトのルート、またはご使用の HATS リッチ・クライアント・プロジェクト用の HATS リッチ・クライアント・ランタイム拡張プロジェクトのルートにあります。**customtables** フォルダーが存在しない場合は、新規に作成します。
2. HATS プロジェクトを開きます。
3. 「接続」ディレクトリーで名前をダブルクリックして、接続を編集します。
4. 「**詳細設定**」タブをクリックします。
5. 「**追加**」をクリックします。
6. 「パラメーターを追加」ダイアログで、「**名前**」のドロップダウン・リストから「**CustomTable**」を選択し、「**値**」フィールドにカスタム・テーブル・ファイルの名前を入力します。
7. 「**OK**」をクリックして、「パラメーターを追加」ダイアログを終了します。
8. 接続設定への変更を保存します。

文字置換の優先順位

HATS で文字を置換する方法はいくつかあります。これらの機能による文字置換は、次の順序で発生します。

1. カスタム・テーブル・サポート
2. UseHodCDRA1399 パラメーター (DBCS の場合のみ)
3. ユーザー定義文字 (UDC) (DBCS の場合のみ)
4. テキスト置換

例えば、カスタム・テーブル・サポートを使用して表示文字を変更し、同じ文字を変更するテキスト置換が定義されている場合、テキスト置換の発生時にその文字は存在しません。

第 23 章 双方向アプリケーション・サポート

この章では、双方向言語でのアプリケーション開発用に HATS により提供される各機能の使用方法について説明します。この内容の大部分は、ヘブライ語とアラビア語のアプリケーション開発者に当てはまります。アラビア語のユーザーに固有の機能は個別に説明します。

HATS では、双方向言語をサポートするために以下の各機能を提供します。

- HATS Toolkit 開発環境の方向に対する制御
- 双方向コード・ページのサポート
- 正確な双方向のテキスト処理と対称スワッピング
- 左から右および右から左の画面カスタマイズのサポート
- 統合オブジェクトから生成されたモデル 1 Web ページ、JSF Web ページ、ボトムアップ従来型 (WSDL ベース) Web サービス、および RESTful Web サービスのサポート
- 標準ポートレットのサポート (標準ポートレット間のポートレット通信のサポートを含む)
- 画面方向の左から右 (LTR) と右から左 (RTL) を切り替える「**画面を反転**」ボタン
- 画面方向に対するさまざまなユーザー制御レベル
- 通常の画面方向と逆方向の場合がある、ホスト・コンポーネント、ウィジェット、およびテキストの方向に対する制御
- デフォルトのレンダリングにおける、右から左および左から右の方向のコンポーネントに対するオプションとしての認識 (拡張レンダリングでの双方向コンポーネントのオプションとしての認識を含む)
- マクロ・プロンプトおよび抽出の方向に対する制御
- ウィジェット設定に使用される双方向混在テキスト
- 各種ブラウザーやプラットフォームでの可視入力フィールドのサポート (Web プロジェクト) および可視テキストのサポート (リッチ・クライアント・プロジェクト)
- グローバル変数の双方向テキストと、画面上での左から右および右から左の方向のテキスト置換
- 3270 セッションにおける右から左への印刷のサポート
- 正しい挿入と抽出を含む、VT 論理と可視双方向セッションのサポート
- 双方向テキストを使用するグローバルな規則
- SWT 端末の双方向サポート
- スプレッドシートの双方向サポート
- リッチ・クライアント・プロジェクトの完全な双方向サポート

この章では、上記のすべての機能について説明します。

© Copyright IBM Corp. 2003, 2015

注: Dojo 変換、ライト・ペン (アテンション)、およびライト・ペン (選択) のホスト・コンポーネントに双方向サポートはありません。

ソフトウェア環境

双方向アプリケーション・サポートの要件を以下に示します。

- 双方向アプリケーションでサポートされるブラウザーは次のとおりです。
  - Firefox V3.4 以降
  - Internet Explorer V6.0 以降
  - Mozilla V1.8 以降
- 正しい双方向データ入力をサポートするには、UTF-8 を指定する必要があります。これを行う方法については、509 ページの『エンコードの設定』を参照してください。
- HATS リッチ・クライアント・アプリケーションのデフォルトの位置合わせは左から右です。例えば、選択リスト、ツールバー・アイコン、およびファンクション・キーは、左から右に位置合わせされます。HATS リッチ・クライアント・プラットフォーム・アプリケーションの位置合わせを右から左にするには、以下のようにする必要があります。
  1. 「新規製品構成」ウィザードを使用し、「**ファイル**」>「**新規**」>「**その他**」>「**プラグイン開発**」>「**製品構成**」を選択して、.product ファイルを作成します。 .product ファイルは、ナビゲーター・ビューの project_name¥src フォルダーに表示されます。
  2. 「**ランチャー**」タブを選択します。
  3. 「プログラムの引数」セクションに `dir rtl` と入力します。

ホスト端末の操作

HATS ホスト端末では Host On-Demand 双方向固有のキー・ストロークが使用可能なため、以下の双方向機能を実行することができます。以下のホスト・ファンクション・キーは、3270、5250 および VT の両方のセッションで使用可能です。

Ctrl+L: ラテン語レイヤー (Latin Layer)
このキー組み合わせにより言語レイヤーがラテン語に変更され、オペレーター情報域 (OIA) は英語を表示するために更新されます。

Ctrl+N: BIDI レイヤー
このキー組み合わせにより、言語レイヤーは双方向に変更され、OIA は双方向言語を表示するために更新されます。

Ctrl+S: 画面を反転
画面方向が左から右の場合、このキー組み合わせにより画面イメージの方向は右から左に変更され、言語レイヤーは BIDI に変更されます。画面方向が右から左の場合は、このキー組み合わせにより画面イメージは逆方向の左から右に変更され、言語レイヤーはラテン語に変更されます。これは、VT 可視セッションでは使用できません。

Ctrl+F: フィールド反転
フィールド方向が左から右の場合、このキー組み合わせによりフィールドの方向が右から左に変更され、カーソルもフィールドの反対側に移動します。

言語レイヤーは、双方向に変更されます。フィールド方向が右から左の場合、このキー組み合わせによりフィールドの方向が左から右に変更され、カーソルもフィールドの反対側に移動します。言語レイヤーはラテン語に変更されます。これは 5250 セッションのみで使用可能です。

以下のホスト・ファンクション・キーは、3270 セッションでのみ使用可能です。

Ctrl+P: プッシュ
フィールドと逆の方向で、テキストを入力および編集できます。

Ctrl+O: プッシュ終了
プッシュ・モードが終了し、カーソルはプッシュ・セグメントの最後に移動します。

Ctrl+A: 自動プッシュ
言語レイヤーを変更して、左から右、および右から左の双方向が混在したテキストを入力できます。

以下のホスト・ファンクション・キーは、5250 セッションでのみ使用可能です。

Ctrl+C: クローズ
1 つのキー・ストローク方向で (左から右、右から左のどちらでも) 入力したデータは、以前に逆の方向で入力したデータと連結されます。カーソルの方向はフィールドの方向と同一に設定され、言語レイヤーはデフォルトでフィールドと同一方向に設定されています。現在の画面方向が左から右である場合、連結されたテキストではカーソルは右側の最初のヌルに置かれます。現在の画面方向が右から左である場合、連結されたテキストではカーソルは左側の最初のヌルに置かれます。

以下のホスト・ファンクション・キーは、Arabic VT セッションでのみ使用可能です。

Ctrl+K: 列見出し
「列見出し」モードによって、テキストの列の間にブランクを入れて挿入部分を中断させ、データの列に対する英語のタイトルが正しい位置を維持することができます。

以下のホスト・ファンクション・キーは、Hebrew VT セッションでのみ使用可能です。

Ctrl+D : カーソルの方向
現在のカーソル方向が左から右の場合は、Ctrl+D を押すと右から左に変更され、再度押すと左から右に戻ります。この機能は、「可視」テキスト・タイプにのみ使用できます。カーソル方向を「右から左」に設定しても、カーソルのアドレッシングおよび移動、文字の挿入および削除、行における削除または表示での削除は影響を受けません。以下の機能が、右から左へのカーソル方向設定から影響を受けます。

- バックスペース (カーソルは右方向へ 1 つの位置だけ移動します)
- 復帰 (カーソルは現在行の右端位置に移動します)
- 改行 (カーソルは次の行の右端位置に移動します)
- 自動折り返しモードでの入力 (現在行は次行の右端位置から継続されます)

Ctrl+D : 文字セット・モード

この機能により、文字セットが 7 ビットと 8 ビット間で切り替わります。現在の文字セットが DEC Hebrew (8 ビット) または ISO Hebrew Supplemental (8 ビット) である場合には、Ctrl+B を押すと、現在の文字セットが Hebrew NRCS (7 ビット) に変更されます。現在の文字セットがヘブライ語 NRCS (7 ビット) である場合には、Ctrl+B を押すと、2 つの 8 ビット文字セットのいずれかが次の規則に基づいてロードされます。

- セッションが 8 ビット文字セットの 1 つを使用するように構成されている場合には、そのセッションの 8 ビット文字セットがロードされます。セッションが 7 ビット文字セットを使用するように構成されている場合には、ISO Hebrew Supplemental がロードされます。現在の言語レイヤーとカーソル方向は変更されません。

ホスト端末に関する制限

ホスト端末では Unicode フィールドがサポートされません。そのため、Unicode フィールドで言語を切り替えることはできません。

双方向のテキストと数字を含む複雑なテキストを画面認識基準に使用しないでください。

画面の取り込み

双方向のセッションでは、画面は左から右、右から左のどちらの方向でも取り込めます。取り込まれた画面は、取り込まれた状態のまま表示されます。画面を右から左の画面として取り込むには、左から右の画面で **Ctrl+S** (画面を反転) を押してから、「**画面キャプチャーを作成**」をクリックします。

双方向ホスト・コンポーネントの認識

通常、ホスト・コンポーネントの方向は画面方向と同じです。ただし、ホスト・コンポーネントの方向が通常の画面方向と逆である場合もあります。HATS プロジェクトで双方向コード・ページを使用する場合、「**画面と逆方向のコンポーネントを認識**」というラベルが付いたチェック・ボックスが「ホスト・コンポーネントを挿入」ウィザードと「画面組み合わせを作成」ウィザードに追加されます。変換にホスト・コンポーネントの追加時、ホスト・コンポーネントの方向が画面方向と逆である場合はこのボックスを選択してください。このボックスを選択すると、HATS は、画面方向と逆の方向であるコマンド・プロンプト、ファンクション・キー、選択リスト、サブファイル、およびヒント付き入力フィールドを認識します。

ファンクション・キーと選択リスト設定では、混在する双方向テキストが使用されます。これはつまり、BIDI1 = BIDI2 のようなストリングは、BIDI1 と BIDI2 が双方向テキストのセグメントで、ファンクション・キーおよび選択リストとして認識されることを意味します。

デフォルトでは、画面のデフォルト・レンダリングでは、(英語セッションと同様に) LTR (左から右) コンポーネントのみが識別されます。ただし、プロジェクト設定エディターで「**レンダリング**」タブの「**デフォルト・レンダリング**」セクションで「**画面と逆方向のコンポーネントを認識**」プロパティーを選択すると、デフォルト・レンダリングで左右両方向のコンポーネントが認識されます。

また、デフォルトでは画面変換のコンポーネントは画面方向に基づいてのみ認識されます。プロジェクト・レベルでは、コンポーネントが認識されない場合に、HATS が逆の画面方向を使用してコンポーネントの認識を試行することを指定できます。これを指定するには、プロジェクト・エディターの「**レンダリング**」タブの「**デフォルト・レンダリング**」の下の「**拡張**」セクションで、「**画面と逆方向のコンポーネントを認識**」チェック・ボックスを選択します。このように選択してもまだコンポーネントが認識されない場合は、「**コンポーネント・レンダリングの失敗時にデフォルト・レンダリングを使用**」設定での指定内容に従ってレンダリングが実行されます。

この設定は、コンポーネント・レベルでも指定できます。そのためには、「レンダリング・オプション」パネルで、「ホスト・コンポーネントを挿入」ウィザードを使用して、変換にコンポーネントを追加する場合に、「**レンダリングの拡張設定**」ボタンをクリックして、「**画面と逆方向のコンポーネントを認識**」ボックスを選択します。コンポーネント・レベルでのこの設定の値は、プロジェクト設定から継承されません。以下のテーブルは、プロジェクト・レベルおよびコンポーネント・レベルでのこの設定の可能な組み合わせと使用される設定を示しています。一般情報については、114 ページの『拡張レンダリング』を参照してください。

表 16. プロジェクトとコンポーネントの設定の組み合わせ

プロジェクト設定	コンポーネント設定	使用される設定
クリア	クリア	コンポーネント
クリア	選択	コンポーネント
選択	クリア	プロジェクト
選択	選択	コンポーネント

ウィジェット方向の制御

アプリケーションが WebSphere Application Server で稼働する場合、ウィジェット表示の制御を可能にする特別なチェック・ボックスが 2 つあります。この制御は、以下に適用されます。

- 新規変換ウィザードの「**デフォルト・レンダリングを使用して事前設定**」または「**フィールドを使用して事前設定**」の場合は、カスタマイズされた画面全体
- 以下の場合は、選択されている特定のウィジェット
  - 「ホスト・コンポーネントを挿入」ウィザード
  - 変換の HATS 固有コンポーネントの「**プロパティー**」ビュー
  - HATS 画面組み合わせの「**レンダリング・オプション**」または「**レンダリング**」タブ

この 2 つのチェック・ボックスは以下のとおりです。

- **ウィジェットでテキストを反転**
- **ウィジェットの方向を反転させる**

「**ウィジェットの方向を反転させる**」チェック・ボックスが選択されている場合、左から右への画面では、GUI イメージの方向が右から左への方向に変更されます。

同様に、「**ウィジェットでテキストを反転**」のチェック・ボックスが選択されている場合、GUI イメージ内のテキストは RTL (右から左) に変更されます。

例えば、以下のテキストを表示している左から右の画面があるとします (大文字は双方向言語データで、小文字は英語データのままです)。

```
BIDI TEXT

pf01=help
```

この画面を RTL としてカスタマイズすると、次のように表示されます。

```
TXET IDIB

pleh=10fp
```

ユーザーの観点から見ると、ファンクション・キー以外はすべて正しく表示されています。ファンクション・キーは認識されてはいますが、逆方向に表示されています。

カスタマイズした画面が正しく表示されるようにするには、カスタマイズした画面にファンクション・キー・ホスト・コンポーネントを挿入するときに、「**ウィジェットの方向を反転させる**」および「**ウィジェットでテキストを反転**」の両方のチェック・ボックスを選択する必要があります。

これを実行すると、画面は以下のように表示されます。

```
TXET IDIB

pf01=help
```

双方向性プロジェクトにおいて、テキスト入力フィールドの代りにドロップダウン・リストを使用すると、ドロップダウン・リスト項目の表題は、画面方向の影響を受けません。ウィジェットの「**ウィジェットでテキストを反転**」チェック・ボックスを選択していなければ、表題は常に可視 LTR フォーマットで表示されます。選択されている場合、表題は可視 RTL フォーマットで表示されます。

また、双方向性 HATS アプリケーションの作成時、保護フィールドから可視入力フィールドに複数行のテキストのコピー、貼り付けを行うと、テキストが反転する場合があります。一度に 1 行のコピー、貼り付けを行うと、テキストの方向は維持されます。

ウィジェットの位置合わせの制御

HATS コンポーネント要素を含んだウィジェットに適切な位置合わせを割り当てるために、以下の HATS 構成パネルにある「**コンポーネントの位置合わせ**」設定を使用できます。

- Web アプリケーションのデフォルト・レンダリングを設定する、プロジェクト設定エディターの「レンダリング」タブ
- 「変換の作成」ウィザードの「レンダリング・オプション」パネル
- 「画面カスタマイズの作成」ウィザードの「アクション」パネル

さらに、個々のウィジェットに適切な位置合わせを割り当てるために、フィールド・ウィジェットおよびテーブル・ウィジェットに対して、「レンダリング・オプ

ション」ウィザードのウィジェット設定ダイアログにある「**フィールドの位置合わせ**」設定および「**テーブルの位置合わせ**」設定も使用できます。

これらの設定の値は、`left` および `right` です。プロジェクト設定内の「**コンポーネントの位置合わせ (Component Alignment)**」設定は、プロジェクト内の全ウィジェットの位置合わせに影響します。プロジェクト設定の値セットは、「変換の作成」ウィザードと「画面カスタマイズの作成」ウィザードのデフォルトになります。個別ウィジェットに指定した位置合わせによって、デフォルト・レンダリングに指定した値がオーバーライドされます。

カスタマイズされたコンポーネントとウィジェットの双方向サポート

Web プロジェクトなどで新しいコンポーネントとウィジェットをカスタマイズする場合は、次のように双方向の対の片方を使用する必要があります。

```
import com.ibm.hats.transform.widgets.BIDI.LabelWidgetBIDI

public class newBIDILabel extends LabelWidgetBIDI implements HTMLRenderer
```

グローバル変数

グローバル変数が抽出される際、抽出されたデータは見ている画面の方向も含めて、画面に表示されたとおりのものです。画面イベントのアクションとして画面に挿入されているグローバル変数は、画面方向に応じて挿入されています。

Web アプリケーションとリッチ・クライアント・アプリケーションの両方において、静的テキストおよび入力フィールドからの双方向データは、Rational SDP 環境およびランタイム環境の両方で、可視形式で、グローバル変数に保管されます。

「グローバル変数の挿入」ウィザードを使用して、グローバル変数値を静的テキストとして双方向性画面の変換に挿入する場合、「**グローバル変数を可視形式で挿入**」というオプションがあります。グローバル変数値を静的テキストとして可視形式で挿入するには、このオプションを選択します。値を静的テキストとして論理形式で挿入するには、このオプションをクリアします。

グローバル変数が、論理 VT セッションから抽出される場合、抽出されるデータは、可視形式に変換されます。VT 論理セッションに挿入されたグローバル変数は、論理形式に変換されます。テキストの方向は画面方向に対応して設定されます。

テキスト置換

プロジェクト・レベルまたは HATS コンポーネント・タグ・レベルの双方向セッションでテキスト置換を使用する場合、使用可能なチェック・ボックスがさらに 3 つあります。

LTR 画面で一致
このオプションを選択すると、左から右の表示画面上のテキストに対するテキスト置換が正確に実行されます。

RTL 画面で一致

このオプションを選択すると、右から左の表示画面上のテキストに対するテキスト置換が正確に実行されます。

画面を反転させて比較

このボックスと「**LTR 画面で一致**」チェック・ボックスを選択した場合、反転ストリングは右から左の画面に一致します。左から右の画面上で「**画面を反転**」ボタンをクリックした場合、データは整合します。

このボックスと「**RTL 画面で一致**」チェック・ボックスを選択した場合、反転ストリングは左から右の画面に一致します。右から左の画面上で「**画面を反転**」ボタンをクリックした場合、データは整合します。

「**LTR 画面で一致**」または「**RTL 画面で一致**」のいずれかを選択する必要があります。

例えば、左から右のホスト画面にテキスト `NO on` が含まれていたと仮定します。テキスト置換で、`no` から `yes` への変換と大/小文字の区別なしを定義すると、選択したチェック・ボックスにより、結果は以下のように異なります。

表 17. 双方向テキスト置換のオプションと結果

選択されたオプション	LTR 画面	RTL 画面
LTR 画面のみ一致	yes on	no ON
RTL 画面のみ一致	NO on	yes ON
LTR 画面と RTL 画面に一致	yes on	yes ON
LTR 画面と反転画面に一致	yes on	no sey
RTL 画面と反転画面に一致	NO sey	yes ON
LTR と RTL の両画面と反転画面に一致	yes sey	yes sey

グローバル規則

双方向セッションでプロジェクト・レベルまたは画面レベルのグローバル規則を使用する場合、(テキスト置換の場合と同様に) さらに 3 つのチェック・ボックスを選択できます。

LTR 画面で一致

このオプションを選択すると、左から右の表示画面上のテキストに対して、設定されたグローバル規則が正確に実行されます。

RTL 画面で一致

このオプションを選択すると、右から左の表示画面上のテキストに対して、設定されたグローバル規則が正確に実行されます。

画面を反転させて比較

このボックスと「**LTR 画面で一致**」チェック・ボックスを選択した場合、反転ストリングは右から左の画面に一致します。左から右の画面上で「**画面を反転**」ボタンをクリックした場合、データは整合します。

このボックスと「**RTL 画面で一致**」チェック・ボックスを選択した場合、反転ストリングは左から右の画面に一致します。右から左の画面上で「**画面を反転**」ボタンをクリックした場合、データは整合します。

詳しくは、 524 ページの表 17を参照してください。

ユーザーによる画面方向反転の可能化

アプリケーション・キーパッド設定ではユーザーの便宜のために、双方向変換で「**画面を反転**」ボタンを提供できます。このボタンは、アプリケーション・キーパッド (Web アプリケーション) と変換ビュー・ツールバー (リッチ・クライアント・アプリケーション) に表示されます。 このボタンをクリックすると、画面の方向が切り替わります。これは、ホスト・システムでは Alt+Enter キーを押して実行していた操作です。

プロジェクト・レベルでは、プロジェクト・エディターの「**レンダリング**」タブの「アプリケーション・キーパッド」セクションに「**画面を反転**」ボタンを組み込むことができます。変換レベルでは、変換エディターの「**アプリケーション・キーパッドを挿入**」機能を使用して「**画面を反転**」ボタンを組み込むことができます。

新しい HATS プロジェクトを作成して双方向コード・ページを選択すると、「接続設定」パネルに「**カスタマイズしていない画面の反転を使用可能にする**」と「**カスタマイズした画面の反転を使用可能にする**」という 2 つのチェック・ボックスが追加されます。これらのチェック・ボックスを使用して、組み込んだ「**画面を反転**」ボタンがこれらの画面に表示されるかどうかを決定します。デフォルトでは、どちらのボックスも選択されています。1 番目のボックスを選択すると、2 番目のボックスが選択可能になります。 2 番目のボックスのみを選択することはできません。

カスタマイズされていない画面とは、いずれの画面カスタマイズにも一致しない画面です。カスタマイズされていない画面がユーザーに対して表示される場合、画面の方向は直前の画面と同じです。最初の画面が、カスタマイズされていない画面の場合、その画面方向は左から右への画面方向になります。直前の画面で画面の方向が変更された場合は、次のカスタマイズされていない画面にはこの変更が継承され、次のカスタマイズされた画面ではリセットされます。

カスタマイズされた画面の初期画面方向は、画面がカスタマイズされた場合と同一の方向です。カスタマイズされている場合も、カスタマイズされていない場合でも、「**画面を反転**」をクリックすると画面方向が変更されます。画面認識の間、反転された画面は、反転される前の同一画面とは別の画面として認識されます。そのため、「**画面を反転**」ボタンを使用すると、画面が認識されない場合もあります。カスタマイズ画面上に表示されるすべてのホスト・コンポーネントの方向が適切であると開発者が確信している場合、その画面の「**画面を反転**」ボタンは使用可能にする必要はありません。カスタマイズ画面の「**画面を反転**」ボタンは使用不可にすることをお勧めします。

アラビア語アプリケーションのユーザーに対して「**画面を反転**」ボタンを使用可能にすると、別のチェック・ボックスで成形の方向を制御できるようになります。詳しくは、 534 ページの『アラビア語コード・ページの機能』を参照してください。

VT 双方向表示オプション

VT 双方向セッションの場合、双方向ユーザーに対して正常に動作するように、端末に一部の特定の双方向プロパティーを設定する必要があります。VT 「**BIDI 表示オプション**」ボタンが、以下のオプションにアクセスするために、「**接続設定**」パネルに追加、表示されます。

テキスト・タイプ (ヘブライ語)
保管されるテキスト文字のフォーマットを決定します。「**可視**」または「**論理**」を選択してください。デフォルトは「**可視**」です。

カーソルの方向 (ヘブライ語可視)
カーソルの方向を左から右 (LTR) または右から左 (RTL) に設定します。カーソルの方向を RTL に設定すると、デフォルトではカーソルは表示された各文字の左に移動するので、すべての文字は RTL の方向に表示されます。一般的には、カーソルの方向が RTL に設定されていると、RTL 方向の入力データを受け入れるように設計されているアプリケーションだけが正常に動作します。デフォルトは **LTR** です。

スマート配列 (アラビア語/ヘブライ語論理)
異なるテキスト属性をもつ文字のセグメントを別々に配列するかどうかを決定します。デフォルトは「**オフ**」です。

テキスト属性表示 (アラビア語/ヘブライ語論理)
スマート配列がオンのときのみ使用可能です。デフォルトは「**はい**」です。

マクロのプロンプトおよび抽出の双方向オプション

Visual Macro Editor またはホスト端末でマクロ・プロンプト (または抽出) を編集する場合は、実行時のマクロ再生中にプロンプト (または抽出) で使用するホスト画面方向を定義できます。オプションは、「**ホスト画面キャプチャーから継承**」、「**LTR**」、および「**RTL**」です。「**ホスト画面キャプチャーから継承**」を選択すると、マクロ・プロンプト (または抽出) に関連付けられたホスト画面に対して定義されたものと同じ画面方向値が使用されます。ホスト画面が関連付けられていない場合は、このオプションは使用できません。「**LTR**」または「**RTL**」のいずれかを指定する必要があります。

HATS V8.0 より前のマクロを HATS V9.5 プロジェクトにコピーする際、または HATS V8.0 より前のプロジェクトを HATS V9.5 にマイグレーションする際に、ホスト画面がマクロ・プロンプト (または抽出) に関連付けられている場合は、「**ホスト画面キャプチャーから継承**」オプションが使用されます。それ以外の場合、デフォルト値である「**LTR**」が使用されます。

ヘブライ語 VT 双方向論理セッションについては、2 つのカーソル方向ラジオ・ボタン (「**左から右**」および「**右から左**」) があります。これらのボタンは、マクロ記録時にプロンプト・アクションに対して入力制御方向を制御します。 VT 接続用のエディターでは、「**BIDI 表示オプション**」をクリックして、これらのラジオ・ボタンを表示してください。

デフォルトの入力制御方向は、LTR 方向です。つまり、プロンプト値は、暗黙的な LTR 形式で表示されます。

マクロ SQL 照会の双方向オプション

双方向言語 (アラビア語とヘブライ語) 用の SQL ウィザードでは、特定の双方向プロパティーを設定する必要があります。これらの設定にアクセスするには、「**拡張マクロ**」エディター・パネルの「**拡張**」ボタンをクリックします。このパネルは、使用中のワークステーションで双方向言語を構成している場合だけに表示されます。以下の設定を使用できます。

ホスト・ファイル・タイプ (Host-File Type)
このオプションは、ホスト・ファイルを論理形式と可視形式のどちらで保存するかを指定します。デフォルトは「**可視**」です。(このオプションは、Java 用 AS/400 ツールボックスから JDBC ドライバーに対して使用することはできません。)

ローカル・ファイル・タイプ (Local-File Type)
このオプションでは、ローカル・ファイルを論理形式と可視形式のどちらにするかを指定します。デフォルトは「**論理**」です。

ホスト・ファイルの方向 (Host-File Orientation)
ホスト・ファイルを LTR 形式と RTL 形式のどちらで保存するかを指定します。デフォルトは「**左から右 (Left-to-Right)**」です。(このオプションは、Java 用 AS/400 ツールボックスから JDBC ドライバーに対して使用することはできません。)

ローカル・ファイルの方向
このオプションでは、ローカル・ファイルを LTR 形式と RTL 形式のどちらにするかを指定します。デフォルトは「**左から右 (Left-to-Right)**」です。

ラーム/アリフ拡張 (Lam-Alef Expansion)
ラーム/アリフ文字の動作を指定します。 SQL ウィザードのステートメントによってホストからアラビア語のデータを受信する場合、「ラームとアリフ」文字が 2 つの文字に拡張されます (「ラームとアリフ」文字の後にスペースがある場合は、「ラーム」の後に「アリフ」が続きます)。

ラーム/アリフ圧縮 (Lam-Alef Compression)
ラーム/アリフ文字の動作を指定します。 SQL ウィザードのステートメントによってホストにアラビア語のデータを送信する場合、「ラーム」とその後に続く「アリフ」が 1 つの文字に圧縮され、「ラームとアリフ」文字の後にスペースが追加されます。このオプションは、アラビア語のシステムだけで使用可能になります。デフォルトは「**オン**」です。(このオプションは、Java 用 AS/400 ツールボックスから JDBC ドライバーに対して使用することはできません。) .

対称スワッピング (Symmetric Swapping)
大括弧などの対称文字の動作を指定します。画面を反転させると、反対向きの相手と置換されます。デフォルトは「**オン**」です。(このオプションは、Java 用 AS/400 ツールボックスから JDBC ドライバーに対して使用することはできません。)

ラウンド・トリップ (Round Trip)
数表示が混在する双方向テキストの処理方法を制御します。

数表示形状 (Numerals Shape)
SQL ウィザードのステートメントで、ホスト・ファイルの数表示の形状を

指定します。数表示形状は、NOMINAL、NATIONAL、および CONTEXTUAL のいずれかです。このオプションは、アラビア語のシステムでのみ使用可能です。デフォルトは **NOMINAL** です。(このオプションは、Java 用 AS/400 ツールボックスから JDBC ドライバーに対して使用することはできません。)

右から左の印刷サポート

双方向ファイルは、RTL ファイルまたは LTR ファイルのいずれかです。RTL データを正しく印刷するには、双方向 3270E デフォルト接続の接続エディターの「**印刷**」タブに表示される「**RTL 印刷を使用可能にする**」 チェック・ボックスを選択する必要があります。PDF ファイルと非 PDF ファイルの両方がサポートされています。

このチェック・ボックスは、印刷が可能な場合に使用可能になります。

アラビア語アプリケーションの場合、RTL 印刷を有効にすると、以下の 2 つのチェック・ボックスが選択可能になります。

- **対称スワッピングを使用可能にする**: 対称スワッピングが使用可能な場合、スワッピング文字は、RTL ファイルの印刷時にスワップされます。
- **数値スワッピングを使用可能にする**: 数値スワッピングが使用可能な場合、RTL ファイルの印刷時に、英語の数表示はアラビア語の数表示、およびアラビア語の数表示は英語の数表示に置換されます。

BMS マップのサポート

BMS マップの双方向言語サポートには、次の内容が含まれます。

1. 双方向コード・ページ : BMS マップ・セットを双方向プロジェクトにインポートする場合は、次のコード・ページを使用する必要があります。
   - アラビア語のプロジェクト : **ホスト・コード・ページ** 420、**BMS ファイル・コード・ページ** 1256 または 864
   - ヘブライ語のプロジェクト : **ホスト・コード・ページ** 424 または 803、**BMS ファイル・コード・ページ** 1255
2. 双方向属性: BMS マップ・セットを双方向プロジェクトにインポートするときには、「**BIDI オプション**」ボタンが表示されます。このボタンをクリックして、以下の BIDI 属性を設定します。
   - インポートされている BMS ソース・ファイルのテキスト・タイプ。
     - テキスト・タイプ: 「**可視**」または「**論理**」
   - BMS マップ・セットから作成される画面キャプチャーに関するオプション:
     - 画面方向: 「**左から右**」または「**右から左**」
     - 対称スワッピング: 「**オン**」または「**オフ**」
     - 数値スワッピング: 「**オン**」または「**オフ**」

統合オブジェクトのサポート

統合オブジェクトのサポートには、抽出したテキストの再配列などを可能にする双方向固有のプロパティーによる統合オブジェクト Bean の拡張が含まれています。詳しくは、「*HATS Web アプリケーション・プログラマーズ・ガイド*」の章『HATS の双方向 API の使用』を参照してください。

統合オブジェクトの Web ページ (モデル 1 および JSF) のサポートには、実行時に抽出/挿入したデータの正しい再配列、および高水準の GUI を使用した、画面および抽出/挿入コントロールの方向に対するアプリケーション開発者による完全な制御が含まれます。

以下の機能がサポートされています。

- モデル 1 Web ページまたは JSF Web ページを作成する場合は、「**モデル 1 Web ページの作成**」または「**JSF Web ページの作成**」ダイアログで「**ページ方向**」ラジオ・ボタン・オプションを使用して、「生成された Web ページの方向を設定」(「**LTR**」または「**RTL**」) することができます。

  「**入力テキスト**」および「**出力テキスト**」ラジオ・ボタン・オプションを使用して「**可視**」または「**論理**」テキスト・インターフェースを選択する機能も提供されています。

  – 入力テキストの「**可視**」インターフェースを選択すると、データが入力フィールド・プロンプトに可視入力フィールド形式で入力されます。出力テキストの「**可視**」インターフェースを選択すると、抽出された結果が可視フィールド形式で表示されます。
  – 「**論理**」出力を選択した場合は、双方向マークを使用する「**両方向を抑制する**」オプションを選択して、双方向混在テキストの正しい再配列を行うことができます。
  – iContext メソッド、または (多くの場合、論理形式になっている) 外部データで HTML コンテンツを変更できるようにするその他のメソッドを使用してマッシュアップを作成するには、「**入力テキスト**」設定で「**論理**」を指定する必要があります。
- 統合オブジェクトの入力 (または出力) プロパティーを定義した場合は、「**入力の定義**」(または「**出力の定義**」) ダイアログに表示されます。プロパティー名、入力 (または出力) コントロールのタイプ、およびコントロール方向が表示されます。これらの設定を変更するには、「**編集**」をクリックします。「**入力プロパティーの定義**」(または「**出力プロパティーの定義**」) ダイアログで、入力 (または出力) コントロールの先行テキストとタイプを変更できます。コントロール方向 (「**LTR**」または「**RTL**」) を変更するには、「**入力コントロール方向**」(または「**出力コントロール方向**」) ラジオ・ボタンを使用します。

  注:

  1. 個々の入力 (または出力) コントロールの方向は、「**ページ方向**」設定とは関係ありません。
  2. それぞれの入力 (または出力) 制御の方向は、デフォルトで、対応するプロンプト (または抽出) と同じ方向に設定されます。
  3. 「**入力コントロール方向**」および「**出力コントロール方向**」ラジオ・ボタンの使用は、Dojo ウィジェット、およびモデル 1 Web ページの HTML コント

ロールでサポートされます。ただし、双方向テキストおよびウィジェット方向は、Dojo コンボ・ボックス・ウィジェットおよびテキスト・ボックス・ウィジェットの「設計」ビューでは正しく表示されません。

注: JSF Web ページを処理している場合、テーブル内の双方向テキストの位置合わせが正しく表示されないことがあります。これは、JSF の設計上の制限です。

ボトムアップ Web サービスのサポート

HATS は、統合オブジェクトから生成されたモデル 1 および JSF の Web ページをサポートするだけでなく、統合オブジェクトから生成されたボトムアップ従来型 (WSDL ベース) Web サービス用の双方向サポートを提供します。この場合、Web サービスは、統合オブジェクトのすべての双方向機能を継承します。この Web サービスは、抽出されたテキストの再配列などを可能にするプロパティーによって拡張されています。これらのプロパティーは、サンプル JSP ページや Web サービス・エクスプローラーなどのツールを使用してテストできます。詳しくは、「*HATS Web アプリケーション・プログラマーズ・ガイド*」 の章『従来の (WSDL ベースの) Web サービスの作成』および『HATS 双方向 API の使用』を参照してください。

HATS Toolkit によって生成されたサンプル JSP ページは論理 LTR アプリケーションであるため、これらのアプリケーションで Web サービスの結果が正しく表示されるのは、プロンプトを呼び出して再配列を抽出し、どちらの場合でも RTL 再配列を「**FALSE**」に設定するように設定している場合のみです。それ以外の場合、結果は正しく表示されません。しかし、プロパティー値の変更が Web サービスにどのように影響するかを確認するために、「誤った」結果を表示することもできます。

サンプル JSP ページで論理 LTR 以外の抽出再配列の正しい結果を表示する必要がある場合、HATS Toolkit によって生成された Result.jsp ファイルを手動で編集する必要がある場合があります。 `<body>` タグがある行の後に、`<div dir=rtl>` (論理 RTL の場合)、`<bdo dir=ltr>` (可視 LTR の場合)、または `<bdo dir=rtl>` (可視 RTL の場合) という行を挿入します。

論理 LTR 以外の再配列の場合に Web サービス・エクスプローラーを使用することはお勧めしません。

Web サービス・サポート・ファイルの作成または更新時に、Web サービスで公開する入出力プロパティーを選択する「プロパティーの選択」ページでは、以下の双方向入出力プロパティーがデフォルトで選択されます。 Web サービスへの公開時に、これらのプロパティーは Web サービスのコンシューマーによって変更できます。

- hExtractRTLTextOrientation
- hExtractReordering
- hPreventBidiRoundTrip
- hPromptRTLTextOrientation
- hPromptReordering

RESTful Web サービスのサポート

RESTful Web サービスは、双方向セッションでサポートされます。 RESTful サービス・リソースの作成または更新時に、選択した HTTP メソッドと統合オブジェクトを反映したデフォルトが設定される「JAX-RS リソース・メソッドの構成」ページで、以下の双方向入出力メソッド・パラメーターもデフォルトに設定されます。

- hPromptReordering (デフォルトは「**論理**」)
- hExtractReordering (デフォルトは「**論理**」)
- hPromptRTLTextOrientation (デフォルトは「**false**」)
- hExtractRTLTextOrientation (デフォルトは「**false**」)
- hPreventBidiRoundTrip (デフォルトは「**false**」)

これらのメソッド・パラメーターの設定は、要件に合うように変更できます。詳しくは、「*HATS Web アプリケーション・プログラマーズ・ガイド*」の章『RESTful Web サービスの作成』および『HATS 双方向 API の使用』を参照してください。

ポートレット・サポート

ポートレット間の通信は、グローバル変数を使用して HATS で実施されます。 HATS グローバル変数は、3270 および 5250 システムでの標準である可視 LTR フォーマットで保管されます。ただし、他のオペレーティング・システム環境で保管されるデータは通常、論理 LTR フォーマットで保管されます。これは、HATS ポートレットはデフォルトで、可視 LTR フォーマットで保管された双方向テキスト・ストリングを使用するのに対して、非 HATS ポートレットは双方向テキスト・ストリングに論理 LTR フォーマットを使用する場合があることを意味します。

HATS ポートレット (可視 LTR フォーマットを使用) と非 HATS ポートレット (論理 LTR フォーマットを使用) との間で双方向テキスト・ストリングを交換できるようにするには、フォーマットの変換が必要であることを HATS に対して指定する必要があります。 HATS ポートレットの送信時にこれを行うには、「グローバル変数の送信」アクションを追加または編集する際に、「**論理 BIDI 形式に変換**」を選択します。詳しくは、187 ページの『「グローバル変数を送信」アクション』を参照してください。 HATS ポートレットの受信時にこれを行うには、受信したプロパティー (JSR168) またはイベント (JSR286) を追加または編集する際に、「**論理 BIDI 形式から変換**」を選択します。詳しくは、148 ページの『ポートレットの設定』を参照してください。

注: 同じフォーマットを使用する 2 つのポートレット間 (例えば、2 つの HATS ポートレット間) で通信を行う際には、変換は実行できません。

ユーザー向けの情報

双方向コード・ページを使用している HATS アプリケーションは、可視入力フィールドと呼ばれる特別な入力フィールドを提供します。論理データの入力および表示をインプリメントする通常のフィールドと異なり、可視フィールドは可視データの入力および表示をインプリメントします。可視フィールドにデータを入力する際、以下の機能を使用できます。

Alt+Shift: 言語選択

このキー組み合わせにより、言語レイヤーがラテン語と双方向言語に切り替わります。

Alt+Enter: 画面を反転

このキー組み合わせにより、画面の方向が反転します。

Shift+NumLock: プッシュ

フィールドと反対方向のテキストを入力および編集することができます。

Shift+NumPad 「/」文字: プッシュ終了

プッシュ・モードが終了し、カーソルはプッシュ・セグメントの最後に移動します。

Alt+NumPad 「/」文字: 自動プッシュ

言語レイヤーを変更して、左から右、および右から左の双方向が混在したテキストを入力できます。自動プッシュは、右からの左の方向に数字を入力するときに特に有用です。プッシュおよびプッシュ終了の機能は、入力されたテキストの言語に応じて自動的にアクティブになります。右から左の方向のフィールドでは、数字またはラテン語を入力すると、言語を変更しなくても自動的にプッシュが開始されます。さらにラテン文字または数字を入力すると、プッシュ・モードが継続します。その他の文字が入力されると自動的にプッシュ・モードは終了します。この機能により、プッシュおよびプッシュ終了を使用しなくても組み込み数字やラテン語で双方向のテキストを入力することができます。

左から右のフィールドでは、双方向の文字を入力すると、自動的にプッシュが開始されます。数字またはラテン文字を入力すると、自動的にモードが終了します。ユーザーは、プッシュやプッシュ終了の代わりに言語レイヤー選択を使用して、組み込み双方向言語でラテン語テキストを入力することができます。

最初に「**最初に自動プッシュ・モードを使用可能にする**」設定を使用して自動プッシュ・モードを使用可能または使用不可にできます。プロジェクト・エディターでは、2 とおりの方法でこの設定を行うことができます。「**概要**」タブで現在のテーマへのリンクをクリックし、「**最初に自動プッシュ・モードを使用可能にする**」設定を使用します。「**その他**」タブで「クライアント設定」セクションを選択し、「**最初に自動プッシュ・モードを使用可能にする**」設定を使用します。

注: 統合オブジェクトへのインターフェースとなる JSF Web ページ上の自動プッシュを使用する場合、必要に応じてフィールド・サイズを手動で変更し、デフォルト・ブラウザー入力フィールド・サイズにより許可される文字よりも多くの文字を入力する必要がある場合があります。

フィールド自動反転

Alt+NumPad 5

- フィールド自動反転が使用可能になっていると、英数字フィールドに初めて入った時、そのフィールドの向きは右から左に設定されます (画面の向きが左から右と右から左の両方の場合)。

- フィールド自動反転が使用可能になっていると、数値フィールドに初めて入った時、そのフィールドの向きは左から右に設定されます (画面の向きが左から右と右から左の両方の場合)。
- フィールド自動反転が使用不可になっていると、(数字の場合、英数字の場合にかかわらず) フィールドに初めて入った時、フィールドの向きは画面の向きと常に同じ向きに設定されます。

フィールド自動反転がアクティブにされている場合、パスワード・フィールド以外のすべてのフィールドは、(上記に説明したのと) 同じ動作を行います。3270 セッションの右から左の画面では、フィールド自動反転が、操作のデフォルト・モードです。

フィールド反転

Alt+Numlock

フィールド反転がアクティブになっている場合、現在の (フォーカスの置かれている) フィールドへの入力方向が変更されます。フィールドが空の場合、カーソルが、フィールドの反対側に位置合わせされます。フィールドが空でない場合、カーソル位置は変りません。フィールド反転は、現在のフィールドに限りアクティブになります。フォーカスが外れると、入力方向はその前の値に戻ります。

クリップボードの機能

双方向データを使用して、プラットフォームの論理エディターから HATS の可視フィールドに、または HATS の可視フィールド (静的テキストを含む) からプラットフォームの論理エディターに、クリップボード操作 (コピー、貼り付け、切り取りなど) を行う場合、この処理は、 Windows プラットフォームで実行されます。Windows ユーザーは、クリップボードを使用して、HATS と、論理データを使用する他のアプリケーションの間で、データを交換できます。

Internet Explorer のクリップボード操作は、メニュー・バーの「編集」オプション、またはショートカット・キー (Ctrl+C、Ctrl+X、Ctrl+V) のどちらを使用しても、実行できます。

Windows プラットフォームで Mozilla ブラウザーを使用する場合は、ショートカット・キーを使用して呼び出した場合にのみ、双方向データのクリップボード操作がサポートされます。

オペレーター情報域 (OIA)

双方向セッションには、オペレーター情報域 (OIA) に表示されるセッション関連情報のほかに、画面と入力の方向、アクティブなレイヤー、自動フィールド反転モードの状況、および成形の状況 (アラビア語セッションの場合) が含まれます。

双方向セッション (キーボード) の関連情報は、ホスト・システムで使用されている形式と類似した形式で表示されます。HATS 変換の OIA でこの情報を表示するには、プロジェクト設定エディターの「**レンダリング**」タブで「**オペレーター情報域**」を選択し、次に「**OIA を表示**」チェック・ボックスを選択します。プロジェクト設定エディターの「**OIA を表示**」チェック・ボックスがクリアされている場合

は、Web アプリケーションでは HATS 双方向状況域がステータス・バーに表示されますが、リッチ・クライアント・アプリケーションでは表示されません。

キーボード・レイヤーの自動切り替え

HATS Web プロジェクトでこの機能を使用すると、画面が反転する際に、双方向言語とラテン語間のキーボード・レイヤーの自動切り替えをシミュレートできます。シミュレーションは、キーボードの再マップを使用して実行されます。 これにより、フィールド・タイプ入力方向に応じて、アラビア/ヘブライ文字またはラテン文字が強制的に追加されます。 例えば、反転された画面では、テキストのタイプ入力が、右から左へ行われる場合、アラビア/ヘブライ文字が、実際のキーボード・レイヤー状態にかかわらず、挿入されます。 この機能を使用可能にするには、「プロジェクト設定」エディターの「**概要**」タブで、現在のテーマをクリックし、「**キーボード・レイヤーの自動切り替えを有効にする**」を選択してください。 この設定は、「プロジェクト設定」エディターの「**その他**」タブの「**クライアント設定**」セクションでも選択できます。

HATS リッチ・クライアント・プロジェクトの振る舞いは、この機能が使用可能である場合と常に同様になります。

注: 「パスワード」フィールド、および可視入力フィールドに基づいていない、その他の特別なフィールドは、システム・キーボード設定にしたがって機能します。すなわちキーボードの再マップはこのようなタイプのフィールドに対して実行されません。

IBM i 5250 Unicode サポート

5250 システムを使用する Unicode フィールドに表示される双方向性テキストは、可視入力フィールドを使用するのではなく、(最初のストロング文字に従って) コンテキスト配列方式を使用して表示および編集されます。

アラビア語コード・ページの機能

以下の機能は、アラビア語コード・ページを使用するプロジェクトに限定されるものです。

HATS Toolkit の画面方向

希望する画面方向を使用して HATS Toolkit 開発環境を開始するには、HATS Toolkit の「スタート」メニュー項目を使用します。 HATS Toolkit のアラビア語翻訳を使用できるかどうかは、基礎となる Rational SDP プラットフォームで提供されるアラビア語サポートに依存します。アラビア語翻訳の Rational SDP にインストールした場合は、アラビア語翻訳の HATS Toolkit を開始する「**HATS Toolkit (アラビア語)**」、および英語翻訳の HATS Toolkit を開始する「**HATS Toolkit (英語)**」の 2 つの HATS Toolkit の「スタート」メニュー項目が用意されています。アラビア語がサポートされるかどうかを判別するには、ご使用の Rational SDP プラットフォームを Knowledge Center で検索してください。

HATS Toolkit の「スタート」メニュー項目を使用せずに、HATS Toolkit で使用する方向を指定するには、以下の表に示すように、-nl または -dir Eclipse 開始引数を使用します。

表 18. HATS Toolkit 方向設定

Eclipse 開始引数	**HATS Toolkit 方向**
-nl <BIDI locale> (BIDI 言語パックがインストールされている)	右から左
-nl <BIDI locale> (BIDI 言語パックがインストールされていない)	左から右
-nl <BIDI locale> -dir rtl	右から左
-dir rtl	右から左
-nl <BIDI locale> -dir ltr	左から右
-nl <non-BIDI locale>	左から右
-dir ltr	左から右

例えば、アラビア語の場合は `eclipse.exe -OpenHatsPerspective -nl ar -dir rtl` を使用して HATS Toolkit を開始します。 Eclipse 開始引数が指定されていない場合、HATS Toolkit 方向は、開発ワークステーションのロケールに関係なく、左から右になります。

JSP ページの方向をカスタマイズ

希望どおりの方向を作成するために、テンプレート、変換、または他の JSP ページ (例えば、stop.jsp ページなど) をカスタマイズしなければならない場合があります。これを行うには、属性 `dir=rtl` および `align=rtl` を HTML タグに適宜追加します。例えば、stop.jsp ページの場合、stop.jsp ページのテキストを囲む `<table>` タグに `dir=rtl` を追加できます。

成形の方向

アプリケーション・キーパッド設定で「**画面を反転**」ボタンを使用可能にすると、アラビア語コード・ページ・プロジェクトの場合は「**RTL 整形**」ボタンも表示されます。「**RTL 整形**」ボタンをクリックすると、RTL 成形の状態がオンとオフで切り替わります。RTL 成形の選択を解除にすると、ユーザーが画面方向とは無関係に LTR ファイルを現在編集中であるということが、HATS に対して通知されます。RTL 成形をオンにすると、現在 RTL ファイルが画面方向とは無関係に編集されていることが HATS に対して通知されるため、同じアラビア語成形データを両方の画面方向で正しく送信することができます。LTR ファイルを編集および送信する前に、RTL 整形の選択が解除されていることを確認する必要があります。また、RTL ファイルを編集および送信する前には、RTL 整形が選択されていることを確認します。

アラビア語の選択的形状

場合により、ホストへの分離フォーマットでのデータ送信が必要になります。そのような要件を満たすために、2 つのチェック・ボックスを使用して、アラビア語セッション用の成形不可フィールド・プロパティーが追加されました。

- **パスワード・フィールドのフィールド整形を禁止**

新しい HATS プロジェクトを作成し、アラビア語コード・ページを選択すると、このチェック・ボックスが表示されます。このチェック・ボックスによって、パスワード・フィールドの整形を使用可能または使用不可にするかが決定されます。すなわち、パスワード・フィールドのアラビア語データが、分離フォームまたは整形フォームでホストに送信されるかが決定されます。

- **フィールドの整形を禁止**

  このチェック・ボックスは、「**ホスト・コンポーネントを挿入**」ウィザードにおいて表示されます。それによって、選択領域の入力フィールド内のアラビア語データが、分離フォームまたは整形フォームでホストに送信されるかが決定されます。

対称スワッピングおよび数字スワッピング

これらのオプションは、アラビア語の 3270 セッションでのみ有効です。対称スワッピングが使用可能である場合、スワッピング文字は右から左の画面にスワップインされます。数字スワッピングが使用可能である場合、英語の数表示は右から左の画面ではアラビア語の数表示に置換され、アラビア語の数表示は右から左の画面では英語の数表示に置換されます。これらのパラメーターは、接続エディターの「**拡張**」タブの「**オプションの拡張接続設定を構成してください。**」セクションで設定されています。パラメーターは symmetricSwapEnabled および numericSwapEnabled として識別されます。また、パラメーターの値は true または false のどちらかです。

HATS ホスト端末は、対称パラメーターおよび数値スワッピング・パラメーターの影響も受けます。

アラビア (西洋) 数字の入力を無効にする

新規の HATS プロジェクトを作成してから、または HATS 接続を作成または編集してから、アラビア語コード・ページ 420 を選択する場合、設定「**RTL 画面でのアラビア (西洋) 数字の入力を使用不可にする**」が表示されます。このオプションを選択し、アラビア (西洋) 数字の入力を無効にします。すなわち、RTL 画面でアラビア (インド) 数字のみを入力できるようにします。これを実行して、送信時に、すべての数字がアラビア (西洋) 数字として送信されるようにします。

キーボード・レイヤーの自動切り替え

アラビア語のキーボードには、一部のキーがわずかに異なる、101 キーボード・レイアウトと 102 キーボード・レイアウトの 2 つのレイアウトがあります。キーボード・レイヤーの自動切り替え機能を使用すると、以下のマッピングが実行されます。

- Tatweel 文字は、両方のキーボードで、shift+j にマップされます。
- 「不等号 (より大)」キーと「不等号 (より小)」キーは、両方のキーボードで、「1」キーの左側のキーにマップされます。
- ZAL 文字は、両方のキーボードで、Enter キーの左側のキーにマップされます。

ホスト端末に関する制限

英語のシステム・キーボード・レイアウトとアラビア語キーボード 101 の両方で以下の制約事項が適用されます。

- 英語キーボード・レイアウトとアラビア語レイヤーを使用する場合に、キーボードのキーパッド領域のドット (.) 文字とスラッシュ (/) 文字が正しく表示されません。
- 英語キーボードとアラビア語レイヤーを使用する場合、アラビア語文字 Tatweel は ZAL 文字として入力します。ZAL 文字を入力するには、**shift+J** キーの組み合わせを使用します。

これらの制限は、アラビア語キーボード 102 には見られません。

アラビア語キーボード 102 を使用する場合は、次の制限が発生します。

- 文字 **S** を入力する目的で Shift+s キーを押しても、何も入力されません。
- 文字 **J** を入力する目的で Shift+j キーを押すと、アラビア語文字の Tatweel が入力されます。
- 文字 **L** を入力する目的で Shift+l キーを押すと、バックスラッシュ (¥) が入力されます。

これらの文字を正しく表示するには、キーボード・レイヤーを「英語」に切り替える必要があります。

画面キャプチャー

右から左に取り込まれた画面でのアラビア語セッションの場合、大括弧と数表示はアプリケーションの対称スワッピングおよび数字スワッピングのオプションの影響を受けます。

注: アラビア語のセッションの場合、キャプチャーされた画面でユーロ記号がブロックとして表示されます。

System i Access for Web でのアラビア語印刷セッションを設定

以下は、System i Access for Web 上でアラビア語データを正しく提示するための設定の要約です。

グローバル設定の場合:

- システム値 QCCSID (37)。このシステム値は英語用であり、HTTP、TomCat、およびバックグラウンド・サーバーが、コード・ページ 420 で表現されない POSIX 文字を使用する必要のある場合に用いて、正常に動作を開始することができます。

SCS レポート:

- このレポートのコード・ページは、プリンター・ファイル・パラメーター (CHRID) に反映されているスプール・ファイルの文字セット ID (CHRID) から取得され、また、プリンター・ファイル・パラメーターは、装置記述、次にシステム値から取得されます。システム値 QCCSID が 420 に設定されていない場合にアラビア語の文字を正常に表示するには、以下のステップを実行してください。

1. 文字 ID パラメーター CHRID (235 420) を含む CHGPRTF または OVRPRTF。グラフィック文字セット 235 とコード・ページ 420。
2. スプール・ファイルを再生成します。

AFP レポート:

- 以下のいずれかの方法で、リソース・ライブラリー・リストに、フォントとオーバーレイ・レポートを追加します。
  - システム・ライブラリー・リスト・システム値 QSYSLIBL にライブラリー QFNTAR3 を追加します。
  - プリンター・ファイル・パラメーターのユーザー・リソース・ライブラリー・リスト USRRSCLIBL に、オーバーレイとフォントを含む各リソース・ライブラリーを追加します。このパラメーターは、上記のシステム値設定より優先し、システム・ライブラリー・リストで定義されている残りのライブラリーのチェックを行いません。

  スプール・ファイルを再生成します。

その他の考慮事項

- アラビア数字をウィジェット・プレビューと展開されている HATS アプリケーションで正確に表示するには、地域設定で数字置換を「**なし**」に設定する必要があります。
- 画面認識は、アラビア語の一部でなく全体を使用して実行される必要があります。
- 双方向機能は、WebSphere Studio V5 JSP 設計の双方向データに関する制限のため、変換設計ではサポートされていません。
- アラビア語セッションのパスワード・フィールドでは、双方向 IBM i 入力制限はサポートされません。

HATS ファイルへの追加

プロジェクトでアラビア語のコード・ページが使用される場合、画面イベント (.evnt) ファイルの記述タグには <orientation>true</orientation> タグが追加されます。値 true は、画面が右から左の方向にカスタマイズされることを示しています。その他の場合は、左から右の方向にカスタマイズされます。

双方向コード・ページを使用しているプロジェクトの場合、接続 (.hco) ファイルが、以下の値を取ることができる、<hodconnection> タグの追加 enableScrRev 属性を持ちます。

(blank)
「**画面を反転**」ボタンはどの画面にも表示されません。

NotCustomized
「**画面を反転**」ボタンは画面カスタマイズに一致しない画面にのみ表示されます。

Customized
「**画面を反転**」ボタンはすべての画面に表示されます。

アラビア語コード・ページを使用しているプロジェクトの場合、接続 (.hco) ファイルが、true または false のいずれかの値を取ることができる、<hodconnection> タグの追加 disableFldShp 属性を持ちます。

3270 セッションで印刷が選択されている場合、<classSetttings> タグに追加の printRTLSupport 設定が表示されます。アラビア語 3270 セッションでは、さらに 2 つの設定が対称スワッピングと数値スワッピングに対して、すなわち、printSymSwapSuppot および printNumSwapSupport が表示されます。この 2 つの印刷設定は、機能の使用可能または使用不可にそれぞれ対応する true または false のいずれかの値を持ちます。

プロジェクトで双方向コード・ページが使用される場合、アプリケーション (.hap) ファイルには、<textReplacement> タグ内の <replace> タグの属性が追加されます。

matchLTR
画面方向が左から右の場合、テキストは置換されます。

matchRTL
画面方向が右から左の場合、テキストは置換されます。

matchReverse
画面方向が反転の場合、テキストは置換されます。

第 24 章 2 バイト文字セットのサポート

この章では、2 バイト文字セット (DBCS) をサポートする HATS アプリケーションの開発に固有の機能と考慮事項について説明します。

一般的な機能に加え、HATS には次のような DBCS をサポートするための特定の機能があります。

- データ・タイプの検査
- フィールド長の検査
- コピー/貼り付け操作の防止
- Input Method Editor (IME)
- AutoIME 切り替え
- シフトアウト/シフトインに関する考慮事項

この章では、これらの特定の機能と、HATS の一般的な機能を使用して DBCS をサポートする際の特別な考慮事項について説明します。

DBCS および SBCS フィールドのサポート

HATS は、以下のタイプの DBCS および SBCS フィールドをサポートしています。

表 19. DBCS および SBCS フィールドのサポート

フィールド・タイプ	フィールドに入力できる内容	プラットフォーム
DBCS 混用 (DBCS mix)	SBCS および DBCS 文字 (シフトアウト/シフトイン (SO/SI) 文字付き)	3270
DBCS のみ (G)	DBCS 文字 (SO/SI なし)	3270
英数字のみ	SBCS 文字のみ、DBCS 文字なし	3270
数値のみ	0 から 9 までの数字、コンマ (,)、ピリオド (.)、正符号 (+)、負符号 (-)、およびスペース ( ) のみ	3270
DBCS オープン (O) (DBCS open (O))	SBCS および DBCS 文字 (SO/SI 付き)	5250
DBCS グラフィック (G) (DBCS graphic (G))	DBCS 図形文字 (SO/SI なし)	5250
DBCS のみ (J)	DBCS 文字 (SO/SI 付き)	5250
DBCS いずれか (E)	DBCS (SO/SI 付き) または SBCS characters	5250
英字のみ (X)	文字 A から Z、コンマ (,)、ピリオド (.)、ハイフン (-)、およびスペース ( ) のみ	5250
英数字シフト (A) (Alphanumeric shift (A))	SBCS 文字のみ、DBCS 文字なし	5250

© Copyright IBM Corp. 2003, 2015

表 19. *DBCS* および *SBCS* フィールドのサポート *(続き)*

フィールド・タイプ	フィールドに入力できる内容	プラットフォーム
数字シフト (N)	SBCS 文字のみ、DBCS 文字なし	5250
符号付き数値 (S) (Signed numeric (S))	数字 0 から 9 のみ	5250
数値のみ (Y)	0 から 9 までの数字、コンマ (,)、ピリオド (.)、正符号 (+)、負符号 (-)、およびスペース ( ) のみ	5250
カタカナ・フィールド (W) (Katakana field (W))	SBCS 文字のみ、DBCS 文字なし	5250
キーボード入力使用禁止 (I)	特殊装置 (ライト・ペンなど) 以外のキーボードからの入力なし	5250
数字のみ (D)	数字 0 から 9 のみ	5250
数値のみの文字 (M)	0 から 9 までの数字、コンマ (,)、ピリオド (.)、正符号 (+)、負符号 (-)、およびスペース ( ) のみ	5250

データ・タイプの検査

各種 DBCS フィールドには、許容されているタイプのデータのみが含まれている必要があります。例えば DBCS グラフィック (G) フィールドには、SO/SI 文字のない DBCS 図形文字のみが含まれている必要があります。HATS では、DBCS フィールドに許容されているタイプのデータのみを入力できるようにするため、データ・タイプ検査が実行されます。

フィールド長の検査

ホスト画面のフィールドでは、DBCS 文字の長さは 2 です。GUI ページのフィールドでは、DBCS の長さは 1 です。この HATS 機能は、GUI フィールドに入力される DBCS 文字の長さを検査し、この GUI フィールドに対応するホスト・フィールドに収まらない文字が入力されないようにします。

コピー/貼り付け操作の防止

この HATS 機能では、ユーザーが正しくないタイプのデータをコピーし、フィールドに貼り付けることを防止します。混合データが貼り付けられると、最初の正しくない文字より後のすべての文字がトリムされます。例えばユーザーが `abc&cde` をフィールドに貼り付けるが、このフィールドのタイプでは `&` が正しくない文字である場合には、`abc` のみが貼り付けられます。

注: この機能は、Windows プラットフォームで Internet Explorer を使用する HATS Web アプリケーションと HATS リッチ・クライアント・アプリケーションでのみサポートされます。

Input Method Editor (IME)

Input Method Editor (IME) は、キーボード入力を処理して DBCS ストリングを生成するフロントエンド・プロセッサー (FEP) です。一般に、IME はオペレーティング・システムのコンポーネントですが、独立した 1 つの製品であったり、またはワード・プロセッサーなどの別のアプリケーションの一部であることがあります。

DBCS ユーザーは、標準サイズのキーボードを使用して数千種類の文字を生成できます。一般に、ユーザーは文字シーケンスを入力し、IME を使用してこのシーケンスを 2 バイト文字に変換します。場合によっては同一文字シーケンスに対して複数の可能な変換 (候補) があるため、変換を再試行する必要があることがあります。この変換処理の実行中は、テキストは論理的には IME に属しています。変換処理後は、ユーザーが IME を使用して変換後の 2 バイト文字をアプリケーションにコミットします。HATS では、IME を使用した HATS フィールドへの 2 バイト文字のコミットがサポートされています。

レンダリングされた HATS フィールドの最大長を超える文字をユーザーが IME に入力しようとしたときの動作は、「**DBCS フィールドの最大長の除去 (Eliminate maximum length in DBCS fields)**」設定および「**SBCS フィールドの最大長の除去 (Eliminate maximum length in SBCS fields)**」設定に応じて変わります。「**DBCS フィールドの最大長の除去 (Eliminate maximum length in DBCS fields)**」設定の詳細については、545 ページの『DBCS フィールドの最大長の除去 (Eliminate maximum length in DBCS fields)』を参照してください。「**SBCS フィールドの最大長の除去 (Eliminate maximum length in SBCS fields)**」設定の詳細については、548 ページの『SBCS での最大長の除去』を参照してください。

AutoIME 切り替え

ホスト画面に DBCS フィールドと SBCS フィールドの両方が含まれていることがよくあります。HATS では、カーソルが DBCS フィールドにあると IME が自動的にオンになり、カーソルが SBCS フィールドにあると IME が自動的にオフになります。

注:

1. この機能は、Windows プラットフォームの HATS リッチ・クライアント・アプリケーションと Internet Explorer を使用する HATS Web アプリケーションでのみサポートされます。
2. AutoIME は、DBCS セッションで DBCS 混用 (3270)、DBCS オープン (O)、DBCS いずれか (E)、およびキーボード入力使用禁止 (I) 以外のすべてのフィールドにおいてサポートされます。
3. 文字レベルではなくフィールド・レベルでサポートされます。
4. コンボ・ウィジェットではサポートされません。
5. 日本語 Windows プラットフォームの HATS Web アプリケーションでは、カーソルがカタカナ (W) フィールドに位置している場合は IME が DBCS (全角) ひらがなモードに切り替わります。ユーザーは、DBCS (全角) カタカナ文字を入力した後、ファンクション・キー F8 を使用して半角カタカナ文字に変換でき、またファンクション・キー F10 を使用して英数字に変換できます。

シフトアウト/シフトインに関する考慮事項

DBCS 混用、DBCS オープン (O)、DBCS のみ (J)、および DBCS いずれか (E) の各フィールドでは、「**レンダリング**」タブの「**DBCS**」セクションで「**無保護のシフトイン/シフトアウト文字はスペースとして表示**」設定が選択されていると、GUI で SO/SI 文字がスペースとして表示されます。この設定が選択されていない場合は、SO/SI 文字は GUI には表示されません。

HATS 統合オブジェクトに使用する接続用にこのプロパティーを構成するには、**omitSOSI** Host On-Demand セッション・パラメーターを使用してください。このパラメーターは、値 **true** または **false** を持つことができます。**true** を設定すると、HATS 統合オブジェクトによって接続の SO/SI 文字が削除されます。 **false** (デフォルト) を設定すると、HATS 統合オブジェクトでは、通常のように SO/SI 文字をスペースで置換します。統合オブジェクト接続に **omitSOSI** プロパティーを追加する方法については、161 ページのオプションの拡張接続設定を構成してくださいを参照してください。

その他の考慮事項

このセクションでは、HATS の一般機能を使用して DBCS をサポートするアプリケーションを開発する際に理解しておくべきその他の考慮事項を説明します。

プロジェクト・テーマ設定

DBCS 最大長の除去

DBCS フィールドの最大長の除去 (Eliminate maximum length in DBCS fields)
この設定をテーマ・レベルで使用して、DBCS フィールドの最大長属性を除去するかどうかを選択します。この設定は、プロジェクト設定エディターでも選択できます。この設定の詳細については、プロジェクト設定エディターの「レンダリング」タブにある 545 ページの『DBCS フィールドの最大長の除去 (Eliminate maximum length in DBCS fields)』設定の説明を参照してください。

プロジェクト設定エディター

DBCS プロジェクトのプロジェクト設定エディターで使用できる設定を以下に示します。その他のプロジェクト設定について詳しくは、107 ページの『第 5 章 HATS プロジェクトの変更』を参照してください。

「レンダリング」タブ

DBCS:

「**レンダリング**」タブで、デフォルト接続に DBCS コード・ページが指定されている場合、プロジェクトに以下の DBCS オプションを構成できます。

無保護のシフトアウト/シフトイン文字はスペースとして表示
GUI で DBCS 混用、DBCS オープン (O)、DBCS のみ (J)、および DBCS いずれか (E) フィールドの無保護のシフトアウト/シフトイン文字をスペー

スとして表示するには、このボックスを選択します。これを選択しないと、シフトイン/シフトアウト文字は GUI で表示されません。

G タイプおよび J タイプのフィールドの SBCS 文字から DBCS 文字への自動変換 (Automatically convert SBCS characters to DBCS for G-type and J-type fields) 3270 および 3270E G タイプまたは 5250 G タイプおよび J タイプ・フィールドで、1 バイト文字を 2 バイト文字に自動変換する場合は、このボックスを選択します。

DBCS フィールドの最大長の除去 (Eliminate maximum length in DBCS fields) Web のみ

選択しなかった場合は、DBCS フィールドの最大長を超える文字を IME に入力できません。

選択した場合は、以下のようになります。

- プロジェクト設定エディターの「レンダリング」タブの「クライアント設定」セクションにある「**自動フィールド移動を有効にする**」が選択されている場合は、DBCS フィールドの最大長を超える文字を IME に入力できます。また、ユーザーが IME によって示される候補を選択してこのフィールドに入力した場合は、超過する文字が次のフィールドにカット・アンド・ペーストされます。
- 「**自動フィールド移動を有効にする**」が選択されていない場合は、DBCS フィールドの最大長を超える文字を IME に入力できます。ただし、ユーザーが IME によって示される候補を選択してこのフィールドに入力した場合は、超過する文字は除去され、他のいずれのフィールドにも入力されません。

この設定は、以下のフィールド・タイプでサポートされます。

- 3270
  - DBCS 混用 (DBCS mix)
  - DBCS のみ (G)
- 5250
  - DBCS のみ (J)
  - DBCS グラフィック (G)
  - DBCS いずれか (E)
  - DBCS オープン (O) (DBCS open (O))

注:

1. この設定は、ポートレット・プロジェクトなどの Web プロジェクトの DBCS コード・ページでのみ使用可能です。
2. この設定は Internet Explorer に対してのみサポートされます。

デフォルトの ATOK 入力モードを Roman に設定 RCP のみ

ATOK デフォルト入力モードを Roman に設定するには、このボックスを選択します。選択されていない場合、ATOK デフォルト入力モードは Hanji に設定されます。リッチ・クライアント・プラットフォームのアプリケーションの場合にのみ有効です。

「その他」タブ

「**その他**」タブの「**クライアント設定**」セクションで、以下の DBCS プロジェクトの設定を考慮してください。

自動フィールド進行を使用可能にする:

HATS Web アプリケーションの場合、自動拡張フィーチャーは、現行の文字数とフィールド長に基づいて拡張時間を決定します。「**レンダリング**」タブの「**DBCS**」セクションで「**無保護のシフトイン/シフトアウト文字はスペースとして表示**」設定が選択されていないと、フィールドに DBCS 文字が含まれている場合に、文字数がフィールド制限に達することはありません。したがって、HATS Web アプリケーションの場合、DBCS 環境で自動拡張機能が必要な場合は、「**無保護のシフトイン/シフトアウト文字はスペースとして表示**」設定を選択する必要があります。

IME を使用してフィールドにデータを入力する場合、KBS.js ファイル内の 2 つのテーブルを使用して自動拡張を起動するキーを定義します。デフォルトのトリガー・キーのセットが提供されています。必要な場合、KBS.js ファイル内のテーブルを編集してデフォルトを変更できます。 KBS.js ファイルを編集するには、「**ナビゲーター**」ビューから、*project_name*¥Web Content¥common フォルダーにあるファイルをダブルクリックします。次の例は、2 つのテーブルに定義されているデフォルト・キーを示しています。例えば、ALT + SBCS/DBCS (キー・コード 25) キー組み合わせは、デフォルトのトリガー・キーです。キーを追加するには、必要に応じて、キー・コード (KBS.js ファイルで定義されている変数名または数値のいずれか) およびキー修飾子 (ALT、CTRL、または Shift) を入力します。キーボード上の特定のキーのキー・コード値を判別する方法については、430 ページの『キー・コード値の判別』を参照してください。

```
var IMEkeysForKeyUp = [
//-Keycode----ALT-CTRL-Shift----//
[  CODE_ENTER,  0,  0,  0  ],
[  CODE_M,      0,  1,  0  ],
[  CODE_ENTER,  0,  1,  0  ]
];

var IMEkeysForKeyDown = [
//-Keycode----ALT-CTRL-Shift----//
[  25,          1,  0,  0  ] //ALT + SBCS/DBCS key
];
```

上書きモード (初期):

DBCS 入力では上書きモード (初期) 機能はサポートされていません。

注:

1. ブラウザーに対して入力を行うと、そのブラウザーが、上書き可能かどうかを制御します。上書きが許可されないブラウザー実装環境もあれば、SBCS に関してのみ上書きが許可されるブラウザー実装環境もあります。
2. 上書きモードが使用可能かどうかにかかわらず、1 つの DBCS 文字が、SBCS または DBCS 文字の前に挿入されます。
3. SBCS 文字は、SO/SI 文字を含む 1 つの DBCS 文字を上書きできます。

フォーカスされたテキストをすべて選択する:

この設定を選択した場合、ユーザーが 5250 J フィールドにタブで移動してフィールドがフォーカスされると、フィールド内のすべてのテキストが選択されます。次に、ユーザーが DBCS 文字を入力した後「フィールドの終了」機能を使用してフィールドを終了した場合は、ストリングの最後の文字が削除されることがあります。

この動作を回避するには、「**フォーカスされたテキストをすべて選択する**」ボックスをクリアして、「**無保護のシフトイン/シフトアウト文字はスペースとして表示**」ボックスを選択します。この場合、カーソルはフィールドの最初の入力位置に正確に配置されます。ユーザーは、この位置から入力を開始した後「フィールドの終了」機能を使用してフィールドを終了できます。

注: これらの考慮事項は、Web アプリケーションでの 5250 J フィールドにしか当てはまりません。

画面イベント・エディター

画面認識基準/開始画面

2 バイト (DBCS) 文字でない文字 (半角カタカナなど) を含む画面認識基準は選択しないでください。選択された DBCS 文字は、選択済み領域に完全に含める必要があります。詳細については、175 ページの『画面認識基準または開始画面』を参照してください。

ホスト・コンポーネント

選択リスト

DBCS プロジェクトの選択リスト・コンポーネントに対して使用可能な設定を以下に示します。その他の選択リスト・コンポーネント設定について詳しくは、242 ページの『選択リスト』を参照してください。

アクション番号を変換 (DBCS のみ)
ホストではリスト項目の先行トークンが 2 バイト文字で表示されるにも関わらず、この項目のターゲット入力フィールドには 1 バイト文字で入力される場合、このパラメーターを使用可能にする必要があります。このため、HATS では、DBCS 項目をホストに送信する前に SBCS 項目に変換します。この設定は、ホスト上で半角文字として表示される項目に対しては適用されません。

サブファイル

サブファイル・コンポーネントの「**列の切れ目**」設定を使用する際の考慮事項を以下に示します。その他のサブファイル・コンポーネント設定について詳しくは、244 ページの『サブファイル』を参照してください。

列の切れ目
列の切れ目によって DBCS 文字を分割することはできません。

テーブル

テーブル・コンポーネントの「**列の切れ目**」設定を使用する際の考慮事項を以下に示します。その他のテーブル・コンポーネント設定について詳しくは、252 ページの『テーブル』を参照してください。

列の切れ目

列の切れ目によって DBCS 2 バイト文字を分割することはできません。

ウィジェット

SBCS での最大長の除去

SBCS フィールドの最大長の除去 (Eliminate maximum length in SBCS fields) (DBCS コード・ページのみ)

選択しなかった場合は、SBCS フィールドの最大長を超える文字を IME に入力できません。

選択した場合は、以下のようになります。

- プロジェクト設定エディターの「レンダリング」タブの「クライアント設定」セクションにある「**自動フィールド移動を有効にする**」が選択されている場合は、SBCS フィールドの最大長を超える文字を IME に入力できます。また、ユーザーが IME によって示される候補を選択してこのフィールドに入力した場合は、超過する文字が次のフィールドにカット・アンド・ペーストされます。
- 「**自動フィールド移動を有効にする**」が選択されていない場合は、SBCS フィールドの最大長を超える文字を IME に入力できます。ただし、ユーザーが IME によって示される候補を選択してこのフィールドに入力した場合は、超過する文字は除去され、他のいずれのフィールドにも入力されません。

この設定は、以下のウィジェットでサポートされます。

- カレンダー
- フィールド
- ポップアップ
- サブファイル (チェック・ボックス)
- サブファイル (ドロップダウン)
- サブファイル (ポップアップ)
- テーブル
- テキスト入力

この設定は、以下のフィールド・タイプでサポートされます。

- 3270
  - 英数字
- 5250
  - カタカナ (W)
  - 英数字 (A)
  - 浮動小数点 (F)
  - 数字シフト (N)

注:

1. この設定は、ポートレット・プロジェクトなどの Web プロジェクトの DBCS コード・ページでのみサポートされます。

2. この設定は Internet Explorer V6 および V7 でのみサポートされます。
3. この設定と「**自動フィールド移動を有効にする**」設定の両方を選択した場合は、ユーザーが IME ではなくキーボードを使用して文字を入力すると、上にリストしたサポートされるフィールド・タイプで自動フィールド移動機能が動作しません。

マクロ・サポート

DBCS プロジェクトでマクロ・サポートを使用する際の考慮事項を以下に示します。

マクロの記録

マクロを記録するときには、「**抽出アクションを追加**」を使用してホスト画面から情報を抽出できます。「抽出アクションを追加」の詳細については、389 ページの『抽出アクションを追加』を参照してください。DBCS 文字の抽出時には、以下で説明する内容を検討してください。

ホスト画面の領域をテーブルに抜き出す場合、テーブルの形式を変更できます。例えば、テーブルの 1 つの列を 2 つの列に分割することができます。「**分割**」をクリックすると、選択された列が HATS によって分割されます。列に 2 バイト (DBCS) 文字が含まれる場合、1 つの DBCS 文字が 2 列に分割されることがあります。この場合、DBCS 文字はどちらの列にも表示されません。抽出アクションの編集中にこの問題が発生した場合は、「**右**」ボタンまたは「**左**」ボタンで列を調整してください。分割された DBCS 文字は、1 つの列に入ると再度表示されます。

マクロのエクスポート

DBCS 文字を含むマクロを HOD マクロにエクスポートし、それを Host On-Demand にインポートする場合、DBCS 文字が判読不能になる可能性があります。 この問題を回避するには、UTF-8 エンコードをサポートするメモ帳などのエディターでエクスポートされたマクロを開き、マクロをネイティブ・エンコードで保存してからマクロを Host On-Demand にインポートします。

統合オブジェクトの作成

DBCS プロジェクトでの統合オブジェクトの作成時の考慮事項を以下に示します。詳しくは、 404 ページの『統合オブジェクトの作成』を参照してください。

統合オブジェクトの作成に使用するマクロの名前を指定するときには、2 バイト (全角) 文字は使用できません。

モバイル装置アプリケーションの操作

Internet Explorer Mobile では、DBCS サポート対象フィーチャーの入力フィールドへの変更は、入力フィールドがフォーカスを失った場合、すなわち、blur イベントが発生した場合に行われます。これは、キー・イベントを使用してリアルタイムで変更が行われるデスクトップ・ブラウザーでのサポートとは異なります。

データ・タイプの検査では、最初の文字が正しくないと、それ以後のすべての文字が、入力フィールドがフォーカスを失った場合、すなわち、blur イベントが発生した場合に除去されます。

詳しくは、 53 ページの『モバイル装置用の HATS アプリケーションの作成』を参照してください。

キーボードおよびディスプレイの文字の再マップ

HATS の Host On-Demand カスタム・テーブルのサポートを使用して、コード・ページ変換テーブルをカスタマイズすることで、ユーザーのキーボードおよびディスプレイの文字を再マップすることができます。詳しくは、 514 ページの『キーボードおよびディスプレイの文字の再マップ』を参照してください。

ユーザー定義文字の処理

DBCS 言語は、表音文字ではなく、表意文字から構成されています。 DBCS 言語では何千もの表意文字が使用されるため、現在定義されているコード・ページまたはフォントが、これらの文字の要件を常に満たすわけではありません。 ユーザー定義文字 (UDC) は、コード・ページまたはフォントで定義されていない、必要な文字をサポートするために使用されます。 UDC マッピング・エディターを使用すると、ローカル・コード・ポイントを、ユーザー定義領域 (UDA) から、この文字のために予約される、指定されたホスト・ コード・ポイントに割り当てることができます。

HATS UDC 機能は、3270、3270E、および 5250 接続タイプをサポートしています。 3270、3270E、および 5250 接続で UDC を表示するには、HATS UDC マッピング・サポートを使用し、3270E 接続で、PDF または PDT モードで UDC を印刷するには、HATS フォント・イメージ・サポートを使用します。

表示サポート

3270、3270E、および 5250 接続での UDC 表示サポートを使用可能にするには、次のステップに従います。

1. UDC ウィザードを起動します。
   a. 「HATS Projects」ビューで、DBCS プロジェクトの接続を編集します。
   b. 接続エディターで、「**拡張**」タブをクリックします。
   c. 「拡張」タブで、「**ユーザー定義文字 (UDC) ファイルのセットアップ**」をクリックします。
2. 「デフォルト・マッピング・テーブルの選択」ページで以下を行います。
   a. 「**UDC マッピング・サポートの使用可能化**」を選択します。
   b. 接続に定義されたコード・ページのマッピング・エディターを起動するには、「**UDC マッピング・エディターの起動**」を選択し、「**起動**」をクリックします。 マッピング・エディターを使用して変更を行います。UDC の変更を保存する場合、.gtt ファイル拡張子を持つマッピング・テーブル・ファイルは、<*RationalSDP_install_directory*>¥hats¥UDC¥udctransl¥ ディレクトリーに保管されます。 以下のコード・ページがサポートされます。

表 20. UDC マッピング・エディター・サポート対象コード・ページ

コード・ページ	場所または用途
930	日本語 (カタカナ)
930	日本 (拡張カタカナ)
933	韓国 (拡張)

表 20. UDC マッピング・エディター・サポート対象コード・ページ (続き)

コード・ページ	場所または用途
937	台湾 (拡張中国語 (繁体字))
939	日本語 (拡張ローマ字)
1364	韓国ユーロ
1371	台湾 (中国語 (繁体字)) ユーロ
1388	PRC (中国語簡体字拡張 : GB18030)
1390	日本語 (カタカナ Unicode 拡張)
1399	日本語 (Latin Unicode 拡張)

c. 既存のマッピング・テーブル・ディレクトリー、例えば、Host On-Demand 製品を使用して作成されたマッピング・テーブル・ディレクトリーをインポートするには、「**使用可能な UDC マッピング・テーブル・ディレクトリーのインポート**」を選択し、「**参照**」をクリックします。 有効なマッピング・テーブル・ディレクトリーには、1 つのマッピング索引ファイル (udctables.inx) および複数のマッピング・テーブル・ファイル (*.gtt) が含まれている必要があります。

d. 「デフォルト・マッピング・テーブルの選択」セクションから、デフォルトとして使用するテーブルを選択します。必要に応じて、「**リストを最新表示**」をクリックし、表示されているテーブルのリストを最新表示します。

e. 「**次へ**」をクリックします。

3. 「要約」ページのすべてのセットアップ情報に注意します。変更を行うには、「**戻る**」をクリックします。変更を行わない場合は、「**完了**」をクリックします。

UDC ウィザードで「**完了**」をクリックすると、HATS は、関連した UDC ファイルを該当する HATS プロジェクトにコピーし、プロジェクトの「**UDC_SETTING**」および「**UDC_TABLE_SELECTION**」拡張接続設定を更新します。

印刷サポート

3270E 接続の UDC 印刷サポートを使用可能にするには、以下のステップに従ってください。

1. UDC ウィザードを起動します。
   a. 「HATS Projects」ビューで、DBCS プロジェクトの接続を編集します。
   b. 接続エディターで、「**拡張**」タブをクリックします。
   c. 「拡張」タブで、「**ユーザー定義文字 (UDC) ファイルのセットアップ**」をクリックします。
2. 「デフォルト・マッピング・テーブルの選択」ページで、「**次へ**」をクリックします。
3. 「フォント・イメージ・ファイルの選択」ページで以下を行います。
   a. Windows ユーザー定義フォントを使用可能なフォント・イメージ・ファイルに変換するために、UDC フォント・イメージ変換ツールのドロップダウンから適切な変換ツールを選択して、「**起動**」をクリックします。

注:

1) 日本語 Windows 2000 オペレーティング・システムの場合、「**日本語 Windows 2000**」 変換ツールを選択し、それ以外の場合、「**標準**」変換ツールを選択します。
2) 日本語 Windows 2000 ツールを使用して、Windows 2000 上で UDC を変換する場合、Unicode で定義されているすべての UDC が変換されます。HATS は、ローカル・ワークステーション・コードで定義されている UDC (合計 1880 文字) だけではなく、ホスト UDC を含む、Unicode で定義されている UDC (コード・ページ 930 および 939 では 0x6941-0x89BD、コード・ページ 1390 および 1399 では 0x6941-0x7FFE) も印刷できます。

b. 変換ツールを使用して、「**変換**」をクリックし、フォント・イメージ・ファイルを作成して保存します。ファイルの名前は、以下のテーブルに表示されているオペレーティング・システム・プラットフォームにより異なります。

表 21. フォント・イメージ・ファイル名

プラットフォーム	フォント・イメージ・ファイル名
日本語 Windows	jpn24.fnt
韓国語 Windows	kor24.fnt
中国語 (簡体字) Windows	chs24.fnt
中国語 (繁体字) Windows	cht24.fnt

c. HATS アプリケーションで使用するフォント・イメージ・ファイルを選択するには、「**参照**」をクリックして、変換ツールを使用して作成したファイルを参照します。Host On-Demand 製品を使用して以前に作成したフォント・イメージ・ファイルを使用する場合は、代わりにそのファイルを参照します。

d. 「**次へ**」をクリックします。

4. 「要約」ページのすべてのセットアップ情報に注意します。変更を行うには、「**戻る**」をクリックします。変更を行わない場合は、「**完了**」をクリックします。

「**完了**」をクリックすると、HATS は、関連した UDC ファイルを該当する HATS プロジェクトにコピーします。

Adobe PDF モードでの UDC の印刷

Adobe PDF モードで UDC を印刷するには、接続設定エディターの「印刷」タブで以下の設定を構成します。

1. 「印刷設定の初期化」セクションのドロップダウン・リストから、「**デフォルト印刷 (Adobe PDF 形式)**」を選択します。
2. 「**初期化**」をクリックします。
3. 以下の「名前/値」テーブルのデフォルト値を使用します。

表 22. 「名前/値」テーブルの設定

名前	値
printDestination	false
printMimeType	application/pdf

表 22. 「名前/値」テーブルの設定 (続き)

名前	値
printSaveAsExtension	.pdf
separateFiles	true
useAdobePDF	true
usePDT	false
useWindowsPrinter	false

PDT モードでの UDC の印刷

PDT モードで UDC を印刷するには、接続設定エディターの「印刷」タブで以下の設定を構成します。

1. 「印刷設定の初期化」セクションのドロップダウン・リストから、「**基本テキスト・ファイル (プレーン・テキスト形式)**」を選択します。
2. 「**初期化**」をクリックします。
3. 「名前/値」テーブルで以下の値を設定します。

表 23. 「名前/値」テーブルの設定

名前	値
PDTFile	PDTfile へのパスは、例えば、*MyPDT/PDTFileName*.hodpdt です。 使用する PDT ファイル名についての、以下のテーブルを参照してください。PDT ファイル名は、コード・ページとサポートされる PDT プリンターにより異なります。
printDestination	false
printMimeType	application/octet-stream
printSaveAsExtension	.out
separateFiles	true
useAdobePDF	false
usePDT	true
useWindowsPrinter	false

注: 以下のテーブルは、「名前/値」テーブルの PDTFile 設定に使用する PDT ファイル名を示しています。PDT ファイル名は、コード・ページとサポートされる PDT プリンターにより異なります。Host On-Demand PDT コンパイラーの使用については、 419 ページの『Host On-Demand PDT コンパイラーの使用』を参照してください。

表 24. PDT ファイル名

コード・ページ	サポートされる PDT プリンター	PDT ファイル名
930 日本語 (カタカナ) 930 日本 (拡張カタカナ) 939 日本 (拡張ローマ字) 1390 日本語 (拡張カタカナ Unicode) 1399 日本語 (拡張ローマ字 Unicode)	ASCII テキスト・モード	/pdfpdt/basic_dbcs.hodpdt
	ESC/P 24-J84 に基づくプリンター	/pdfpdt/esc_p.hodpdt
	IBM 5577-B02、F02、G02、H02	/pdfpdt/ibm5577.hodpdt
	IBM 5585-H01 プリンター	/pdfpdt/ibm5585.hodpdt
	IBM 5587-G01、H01 (拡張機能なし)	/pdfpdt/ibm5587.hodpdt
	Lips3a4 プリンター	/pdfpdt/lips3a4.hodpdt
	Lips3b4 プリンター	/pdfpdt/lips3b4.hodpdt
933 韓国 (拡張) 1364 韓国ユーロ	韓国 IBM 5577 プリンター	/pdfpdt/ibm5577k.hodpdt
	Ks_jo プリンター	/pdfpdt/ks_jo.hodpdt
	Ks_wan プリンター	/pdfpdt/ks_wan.hodpdt
	Kssm_jo プリンター	/pdfpdt/kssm_jo.hodpdt
	Kssm_wan プリンター	/pdfpdt/kssm_wan.hodpdt
937 台湾 (拡張中国語 (繁体字)) 1371 台湾 (中国語 (繁体字)) ユーロ	中国語 (繁体字) ESC/P プリンター (5550)	/pdfpdt/esc_5550.hodpdt
	中国語 (繁体字) ESC/P プリンター (big-5)	/pdfpdt/esc_big5.hodpdt
	中国語 (繁体字) ESC/P プリンター (cns)	/pdfpdt/esc_cns.hodpdt
	中国語 (繁体字) ESC/P プリンター (tca)	/pdfpdt/esc_tca.hodpdt
	中国語 (繁体字) IBM 5577 プリンター (拡張機能なし)	/pdfpdt/ibm5577t.hodpdt
	中国語 (繁体字) IBM 5585 プリンター	/pdfpdt/ibm5585t.hodpdt
1388 PRC (拡張中国語 (簡体字); GB18030)	中国語 (簡体字) ESC/P プリンター	/pdfpdt/esc_pp.hodpdt

ローカル PDT プリンターを使用して、PDT モードでファイル・コンテンツを印刷するには、以下のステップに従います。

1. HATS 印刷ジョブ・リストから PDT ファイルをダウンロードします。
2. DOS プロンプトに、`type xxxx.out > prn` と入力し、ファイルをローカル PDT プリンターに送信します。

制限

- UDC 変換ツールの制限

- 中国語 (繁体字) Windows 環境では、IBM の Big5 UDC より 13 個多い UDC があります。 このため、最後の 13 個の UDC が 0xC8F2-0xC8FE の範囲に定義されている場合、これらはユーティリティーにより無視され、使用することはできません。
- 韓国語 Windows 環境では、ローカル・コード・ページ 949 のみがサポートされています。 0xC9A1-0xC0FE および 0xFEA1-0xFEFE の範囲で、188 個の UDC を定義して印刷できます。

• ユーザー定義領域 (UDA) の範囲は、Windows プラットフォームに基づいて定義されるため、 UDC は、Windows プラットフォームのみで表示できます。

付録 A. ランタイム・プロパティー・ファイル

ログ、トレース、およびライセンスの設定などのランタイム設定は、次の 2 つの異なるファイルのいずれかに保管されます。

- runtime.properties ファイルは、ランタイム環境および「サーバーで実行」モード (Web アプリケーションの場合) または「実行」モード (リッチ・クライアント・アプリケーションの場合) で稼働するローカル・テスト環境で、HATS によって使用されます。
- runtime-debug.properties ファイルは、「サーバーでデバッグ」モード (Web アプリケーションの場合) または「デバッグ」モード (リッチ・クライアント・アプリケーションの場合) のローカル・テスト環境で、HATS によって使用されます。

Web アプリケーションのランタイム設定の管理方法について詳しくは、450 ページの『問題判別コンポーネントの管理』を参照してください。リッチ・クライアント・アプリケーションのランタイム設定の管理方法について詳しくは、93 ページの『HATS リッチ・クライアント・アプリケーションの管理』を参照してください。

runtime.properties ファイルと runtime-debug.properties ファイルには、次のプロパティーが含まれています。

adminPortNum

adminPortNum は、管理の有効範囲を変更するときに HATS 管理コンソールにより使用されるデフォルト値です。ポート値は、アクティブな WebSphere BOOTSTRAP_ADDRESS ポートでなければなりません。デフォルト値は 2809 です。これは、WebSphere ベース・インストールの server1 のデフォルト BOOTSTRAP_ADDRESS ポート、または Deployment Manager セルにアプリケーション・サーバー・ノードを統合するときのノード・エージェント・サーバーのデフォルト BOOTSTRAP_ADDRESS ポートです。

ioPatternKey **Web のみ**

トレースする統合オブジェクトを指定する 1 つ以上のパターン。以下の trace.INTEGRATIONOBJECT の説明を参照してください。

各パターンには、1 つ以上のワイルドカード (*) 文字を使用できます。例えば *IntegrationObject.Callup** は、`Callup` で始まる名前を持つすべての統合オブジェクトに対してトレースを使用可能にすることを指定します。すべての統合オブジェクトをトレースするには、*IntegrationObject.** を指定します。

複数のパターンを指定する場合には、各パターンをコンマで区切ってください。

numLicenses

購入したライセンスの数を指定します。HATS は、ホスト・リソースへの HATS 接続の数を追跡し、その値が購入したライセンスの数を超過すると、メッセージをログに記録します。

© Copyright IBM Corp. 2003, 2015

この値は整数です。デフォルト値はありません。「許可ユーザー」ライセンスの有効値は 1 から 50000 です。「バリュー・ユニット」ライセンスでは、値は、-1 でなければなりません。

licenseHardLimit
HATS ライセンス資格を含む暗号化されたフィールド。このフィールドは、手動で変更しないでください。

licenseTracking
HATS でライセンス使用を記録するかどうかを指定します。この値は 2 進数です。デフォルトは **0** です。

0 HATS はライセンス使用を記録しません。

1 HATS は 1 つのノード内のすべてのアプリケーション・サーバーのライセンス使用を記録します。HATS は、ホストまたはデータベース・リソースへの HATS 接続の数を追跡し、その値が購入したライセンスの数を超過すると、メッセージをログに記録します。ライセンス使用情報は、サーバー上の HATS インストール・ディレクトリーの `log` ディレクトリーにある license*x*.txt という名前のファイルに書き込まれます。ここで、*x* は 1 または 2 です。

ライセンス使用状況ファイルの最大サイズは、512 KB です。license1.txt ファイルのサイズが 512 KB に達するか、または HATS サーバーが再始動されると、そのファイル名は license2.txt に変更され、新しい license1.txt ファイルが作成されます。新しい license1.txt ファイルには、最新のライセンス使用状況情報が入ります。新しい license1.txt が 512 KB に達し、名前変更されると、古い license2.txt は削除されます。

ライセンス使用状況ファイルは一連の行から成り、1 行に以下の情報が入っています。各行はそれぞれ 1 時間分の操作に対応しています。値と値の間はスペース ( ) で区切られます。

1. 日付
2. 時刻
3. サーバーが始動されてからの最大ライセンス数
4. 過去 1 時間における最大ライセンス数 (最後の 60 エントリーの最大値)
5. 1 分ごと (1 から 60) のライセンス数

licenseFile
ライセンス使用状況情報のライセンス・ファイルの名前を生成するときにテンプレートとして使用される名前です。このプロパティーのデフォルト値は license.txt です。

logFile ログ・メッセージが書き込まれるアプリケーション・サーバー・ファイルの各セットに名前を生成するためのテンプレートとして使用する名前。ログ・ファイルのデフォルトのベース名は、*messages.txt* です。

logMask
特定のマクロ・エラーが発生すると、ホスト画面のイメージ (画面ダンプ) が HATS ログ・ファイルに入れられます。

画面ダンプは INFO メッセージに分類されます。HATS ログに記録するメッセージのタイプを制御するには、この設定を使用します。例えば、セキュリティー上の懸念事項により HATS ログに画面が表示されるのを避ける場合は、logMask=6 を設定して、INFO メッセージ (画面ダンプを含む) がログに入れられることを防止します。

logMask 設定を使用して ERROR メッセージをシャットオフすることはできません。

4 ERROR メッセージを記録します。

5 ERROR および INFO メッセージを記録します。

6 ERROR および WARNING メッセージを記録します。

7 ERROR、WARNING および INFO メッセージを記録します。

maxLogFiles

メッセージ・ファイルの最大数。デフォルトは **2** です。

runtime.properties の中の基本ログ・ファイル名が、各アプリケーション・サーバーに固有のメッセージ・ログ・ファイルのセットを生成するためのテンプレートとして使用されます。デフォルトの基本ログ・ファイル名は runtime.properties で変更できます。HATS を実行しているアプリケーション・サーバーの名前は、アンダースコアー文字 (_)、HATS インスタンスの名前、アンダースコアー文字 (_) をこの順序で連結したものになります。

アプリケーション・サーバーのログ・ファイルのテンプレートを生成するには、基本ファイル名に HATS インスタンス ID (*_SSS_* など) を付加します。ログ・ファイルの場合、これは *messages _SSS_.txt* になります。さらに、複数のファイルを区別する場合、この名前に索引 (1、2、3 ...) を追加します。そこで、例えば、HATS インスタンス ID が *cell_node_server* の場合、アプリケーション・サーバーのログ・ファイル名は *messages_cell_node_server_.txt* になります。複数のログ・ファイルが構成されている場合、このアプリケーション・サーバーのログ・ファイル名は *messages_cell_node_server_1.txt, messages_cell_node_server_2.txt ...* となります。

messages_cell_node_server_1.txt のサイズが maxLogFileSize に達すると、そのファイルはクローズされ、*messages_cell_node_server_2.txt* に名前変更されます。新しい *messages_cell_node_server_1.txt* がオープンされます。

messages_cell_node_server_1.txt のサイズが再度 maxLogFileSize に達すると、古いログ・ファイルの名前が変更されます。例えば、*messages_cell_node_server_2.txt* は *messages_cell_node_server_3.txt* に変更されます。*messages_cell_node_server_1.txt* は *messages_cell_node_server_2.txt* に名前変更され、新しい *messages_cell_node_server_1.txt* ファイルがオープンされます。

maxLogFiles の数を超過すると、最も古いファイルが削除されます。

注: WebSphere Application Server for z/OS 上で実行している場合、ログ・ファイル名のサーバー部分には、ログへの書き込みを実行するサーバントのアドレス・スペース ID も含まれています (例:

messages_asid_1.txt)。詳細については、 451 ページの『ログ・ファイルとトレース・ファイルの名前』を参照してください。

maxLogFileSize

メッセージ・ログ・ファイルの最大サイズを KB 単位で指定します。このサイズに達すると、新しいログ・ファイルがオープンされます。

この値は整数です。デフォルトは **512** KB です。

traceFile

HATS トレース・メッセージの書き込み先のファイル名を生成するためにテンプレートとして使用する名前。トレース・ファイルのデフォルトの基本名は、*trace.txt* です。

maxTraceFiles

トレース情報ファイルの最大数。デフォルトは「**5**」です。

server.properties の中の基本トレース・ファイル名が、各アプリケーション・サーバーに固有のトレース・ファイルのセットを生成するためのテンプレートとして使用されます。デフォルトの基本トレース・ファイル名は runtime.properties で変更できます。HATS を実行しているアプリケーション・サーバーの名前は、アンダースコアー文字 (_)、HATS サーバー・インスタンスの名前、アンダースコアー文字 (_) をこの順序で連結したものになります。

アプリケーション・サーバーのトレース・ファイルのテンプレートを生成するには、基本ファイル名に HATS サーバー・インスタンス ID (*_SSS_* など) を付加します。このファイル名は *trace _SSS_.txt* となります。さらに、複数のトレース・ファイルを区別する場合、この名前に索引 (1、2、3 ...) を追加します。そこで、例えば、HATS サーバー・インスタンス ID が *cell_node_server* の場合、アプリケーション・サーバーのトレース・ファイル名は *trace_cell_node_server_.txt* になります。複数のトレース・ファイルが構成されている場合、このアプリケーション・サーバーのトレース・ファイル名は *trace_cell_node_server_1.txt, trace_cell_node_server_2.txt ...* となります。

trace_cell_node_server_1.txt のサイズが maxTraceFileSize に達すると、そのファイルはクローズされ、*trace_cell_node_server_2.txt* に名前変更されます。新しい *trace_cell_node_server_1.txt* がオープンされます。

trace_cell_node_server_1.txt のサイズが再度 maxTraceFileSize に達すると、古いトレース・ファイルの名前が変更されます。例えば、*trace_cell_node_server_2.txt* は *trace_cell_node_server_3.txt* に変更されます。*trace_cell_node_server_1.txt* は *trace_cell_node_server_2.txt* に名前変更され、新しい *trace_cell_node_server_1.txt* ファイルがオープンされます。

maxTraceFiles の数を超過すると、最も古いファイルが削除されます。

注: WebSphere Application Server for z/OS 上で実行している場合、トレース・ファイル名のサーバー部分には、ログへの書き込みを実行するサーバントのアドレス・スペース ID も含まれています (例: *trace_asid_1.txt*)。詳細については、 451 ページの『ログ・ファイルとトレース・ファイルの名前』を参照してください。

maxTraceFileSize
トレース・ファイルの最大サイズを KB 単位で指定します。このサイズに達すると、新しいトレース・ファイルがオープンされます。

この値は整数です。デフォルトは **10240** KB です。

traceLogDirectory
アプリケーションの EAR を展開する場合にログとトレース出力を書き込むためのディレクトリーを設定します。このキーワードを指定しない場合、または指定された値が有効でない場合は、使用されるディレクトリーは「*logs*」で、これは EAR のインストール・ディレクトリーの下に配置されます。この値は、EAR がアプリケーション・サーバーに展開された場合のみに使用されます。この値は、Toolkit 環境では無効です。

maxHODThreadManagerThreads
Host On-Demand ThreadManager スレッド・プールに作成できるスレッドの最大数を指定します。10 から 10000 までの値を指定できます。この範囲外の値を指定すると、設定が無視され、デフォルトの Host On-Demand 値である **10000** が使用されます。

トレース・オプション

このセクションで説明するトレース・オプションで使用されるトレース・レベル値は、16 進数字ストリングを表します。数字ストリングの各ビットは、そのランタイムに対する 1 つのトレース・タイプを制御します。デフォルトは以下のとおりです。

```
tracelevel.1 : 0000000000020000
tracelevel.2 : 000000000000020f
tracelevel.3 : 000000000004023f  (minimum)
tracelevel.4 : 0000000000041a3f
tracelevel.5 : 00000000000c1bbf  (normal)
tracelevel.6 : 00000000000c1bbf
tracelevel.7 : 00000000001c1bbf  (maximum)
tracelevel.8 : 00000000001c1bbf
tracelevel.9 : 00000000001c1bbf
```

以下に、トレース・マスク内の各ビットの定義を示します。

```
x000001  - Informational messages
x000002  - Warning messages
x000004  - Error messages
x000008  - Critical error messages
x000010  - API traces
x000020  - Callback API traces
x000080  - Method entry
x000100  - Method exit
x000200  - Exceptions
x000400  - Miscellaneous traces
x000800  - Object creation
x001000  - Object disposal
x020000  - performance tracing - use this value alone
x040000  - Miscellaneous data - level 1
x080000  - Miscellaneous data - level 2
x100000  - Miscellaneous data - level 3
```

トレース・レベル値をカスタマイズするには、トレース・マスクの 16 進値を合計します。例えば、`tracelevel.9=180` と指定し、`trace.RUNTIME=9` を使用すると、メソッドの開始とメソッド出口のランタイム・トレースのみが実行されます。

trace.RUNTIME

メイン・ランタイムのトレース・レベル、およびトレース・レベルを指定していない RUNTIME.* で使用されるすべての設定に関するトレース・レベルを指定します。

この値は、0 から 9 の整数です。デフォルトは **0** です。このデフォルト値ではランタイム・トレースは実行されません。

trace.RUNTIME.ACTION

イベント・アクションのトレースのレベルを指定します。この設定は、trace.RUNTIME の設定をオーバーライドします。

この値は、0 から 9 の整数です。デフォルトは **0** です。このデフォルト値ではイベント・アクション・トレースは実行されません。

trace.TRANSFORM

HATS 変換のトレース・レベル、およびトレース・レベルを指定していない TRANSFORM.* で使用されるすべての設定に関するトレース・レベルを指定します。

この値は、0 から 9 の整数です。デフォルトは **0** です。このデフォルト値では変換トレースは実行されません。

trace.TRANSFORM.COMPONENT

HATS コンポーネントに対するトレースのレベルを指定します。この設定は、trace.TRANSFORM の設定をオーバーライドします。

この値は、0 から 9 の整数です。デフォルトは **0** です。このデフォルト値ではコンポーネント・トレースは実行されません。

trace.TRANSFORM.WIDGET

HATS ウィジェットに対するトレースのレベルを指定します。この設定は、trace.TRANSFORM の設定をオーバーライドします。

この値は、0 から 9 の整数です。デフォルトは **0** です。このデフォルト値ではウィジェット・トレースは実行されません。

trace.INTEGRATIONOBJECT Web のみ

統合オブジェクトに対するトレースのレベルを指定します。

この値は、0 から 9 の整数です。デフォルトは **0** です。このデフォルト値では統合オブジェクトのトレースは実行されません。

trace.UTIL

HATS ランタイム・ユーティリティーに対するトレースのレベルを指定します。

この値は、0 から 9 の整数です。デフォルトは **0** です。このデフォルト値ではランタイム・ユーティリティー・トレースは実行されません。

trace.APPLET Web のみ

アプレットのトレースのレベルを指定します。

値は整数であり、以下のようになります。

0 トレースは実行されません。

3 最小レベルのトレースが実行されます。

5 通常レベルのトレースが実行されます。

デフォルトは **0** です。

Host On-Demand トレース

trace.HOD.COMMEVENT

Host On-Demand COMM イベントのトレース・レベルを指定します。

0 Host On-Demand COMM イベント・トレースは使用不可です。

1 Host On-Demand COMM イベント・トレースが使用可能です。

この値は 2 進数です。デフォルトは **0** です。

trace.HOD.DISPLAYTERMINAL

Host On-Demand が接続ごとに端末ウィンドウを表示するかどうかを指定します。

0 Host On-Demand は接続ごとに端末ウィンドウを表示しません。

1 Host On-Demand は接続ごとに端末ウィンドウを表示します。

この値は 2 進数です。デフォルトは **0** です。

trace.HOD.DISPLAYTERMINAL が 1 の場合、HATS がホスト接続を作成すると (例えばアプリケーションからの要求に応答して作成するなど)、自動的にホスト端末表示が作成されます。このプロパティーが 0 に設定されていると、この処理は行われません。ただし、ホスト接続作成時にホスト端末表示が自動的に作成されるかどうかに関係なく、HATS 管理者は HATS 管理コンソールを使用して各ホスト接続のホスト端末表示をオンまたはオフにできます。

trace.HOD.DS

Host On-Demand データ・ストリーム・トレースのトレース・レベルを指定します。

この値は、0 から 3 の整数です。

0 トレースは実行されません。

1 最小レベルのトレースが実行されます。

2 通常レベルのトレースが実行されます。

3 最大レベルのトレースが実行されます。

デフォルトは **0** です。

trace.HOD.MACRO

Host On-Demand マクロのトレースのレベルを指定します。

この値は、0 から 2 の整数です。

0 Host On-Demand マクロ・トレースは使用不可です。

1 Host On-Demand イベント・トレースが使用可能です。

2 Host On-Demand サポート・トレースが使用可能です。

デフォルトは **0** です。

trace.HOD.PS

Host On-Demand 表示スペースのトレース・レベルを指定します。

この値は、0 から 3 の整数です。

0 トレースは実行されません。

1 最小レベルのトレースが実行されます。

2 通常レベルのトレースが実行されます。

3 最大レベルのトレースが実行されます。

デフォルトは **0** です。

trace.HOD.PSEVENT

Host On-Demand PS イベントのトレース・レベルを指定します。

0 Host On-Demand PS イベント・トレースは使用不可です。

1 Host On-Demand PS イベント・トレースが使用可能です。

この値は 2 進数です。デフォルトは **0** です。

trace.HOD.OIAEVENT

Host On-Demand OIA イベントのトレース・レベルを指定します。

0 Host On-Demand OIA イベント・トレースは使用不可です。

1 Host On-Demand OIA イベント・トレースが使用可能です。

この値は 2 進数です。デフォルトは **0** です。

trace.HOD.SESSION

Host On-Demand セッションのトレースのレベルを指定します。

この値は、0 から 3 の整数です。

0 トレースは実行されません。

1 最小レベルのトレースが実行されます。

2 通常レベルのトレースが実行されます。

3 最大レベルのトレースが実行されます。

デフォルトは **0** です。

trace.HOD.TRANSPORT

Host On-Demand トランスポートのトレース・レベルを指定します。

この値は、0 から 3 の整数です。

0 トレースは実行されません。

1 最小レベルのトレースが実行されます。

2 通常レベルのトレースが実行されます。

3 最大レベルのトレースが実行されます。

デフォルトは **0** です。

trace.HOD.USERMACRO

Host On-Demand ユーザー・マクロでのトレース・アクションのトレース・レベルを指定します。

この値は、0 から 3 の整数です。

0 トレースは実行されません。

1 最小レベルのトレースが実行されます。

2 通常レベルのトレースが実行されます。

3 最大レベルのトレースが実行されます。

デフォルトは **0** です。

ホスト・シミュレーション・トレース

recordSimulationTrace

ホスト・シミュレーション・トレースの記録を実行時に使用可能にするかどうかを指定します。

0 ホスト・シミュレーション・トレースの記録は使用可能ではありません。

1 ホスト・シミュレーション・トレースの記録は使用可能です。

この値は 2 進数です。デフォルトは **0** です。

startPort

実行時に、複数ホスト・セッションのホスト・シミュレーション・トレース記録を同時に実行するために使用するポート範囲の開始ポートを指定します。デフォルトは、ポート **7021** です。

endPort

実行時に、複数ホスト・セッションのホスト・シミュレーション・トレース記録を同時に実行するために使用するポート範囲の終了ポートを指定します。デフォルトは、ポートは **7050** です。

付録 B. HATS 画面確定のリファレンス

ここでは、画面確定の概要、画面確定の設定、および画面確定に影響する他の設定について説明します。

画面確定の概要

ホスト・アプリケーションは、アウトバウンド・データをいくつのパケットに分けて伝送する場合があります。ただし、この数はクライアントに通知されません。この状態は、端末またはヘビー・クライアント端末エミュレーターを使用すると、簡単に処理できます。これは、端末またはヘビー・クライアント端末エミュレーターでは、通信が常にアクティブで、必要な場合にクライアントのビューがいつでも更新されるためです。ただし、HATS を使用して画面ベースのホスト・アプリケーションを HTML 形式に変換する場合、クライアントが (アプリケーション・サーバーの HATS ランタイムを通じて間接的に) ホストに接続されるのは、クライアントから要求されたブラウザーの更新が処理されている間だけです。そのため、HATS は、このブラウザー更新サイクルで受信したアウトバウンド・データを分析し、現在のホスト画面を変換などのイベント処理のために画面認識エンジンに送信するタイミングを判断する必要があります。アウトバウンド・データを分析して、現在のホスト画面を呼び出すタイミングを決定するこの処理を、画面確定と呼びます。

画面確定は、以下の場合に HATS ランタイムで実行されます。

- 画面を変換する場合のみ (つまり、次の画面を確認するための独自の条件を含むマクロまたは統合オブジェクトを実行していない場合)
- ホストとの対話 ([Enter] などの AID キーによるものなど) が発生し、アウトバウンド・データが送信される可能性がある場合

画面が確定されると、HATS では、その後の処理のために、ホスト画面 (表示スペース) が調べられ、使用可能な画面カスタマイズ (画面認識) のセットと比較されます。確定された画面が、意図された画面であることが重要です。

画面確定手順

HATS では、画面確定が主に以下の 2 つの操作で実行されます。

- アウトバウンド・データの分析
- オペレーター情報域 (OIA) フラグの取得待ち

HATS では、アウトバウンド・データの分析時に、複数のアウトバウンド・データ分析計画のいずれかが実行され、ホスト・アプリケーションのデータ送信が完了するタイミングが判断されます。これらの各計画は、それぞれ異なる事例で使用され、動作をカスタマイズするための設定がそれぞれに含まれています。

OIA フラグの取得を待機する場合、アウトバンド・データの分析に使用される計画に依存しません。そのため、セッションの OIA フラグが検査され、ホストの準備が整い、ユーザーがインバウンド・データを送信できる状態であることが確認されます。この動作をカスタマイズする設定もあります。

© Copyright IBM Corp. 2003, 2015

アウトバウンド・データの分析

実行時の事情に応じて、HATS では、使用可能な複数の計画の 1 つを使用して、ホストが各画面更新の送信を完了したかを判断します。 HATS では複数の要因に基づいた計画を使用します。要因には、ホストの接続タイプ、接続状態、ユーザー構成可能な各種設定などが含まれます。

接続が使用中の計画を確認するには、HATS ランタイムをトレースします。トレースの詳細については、 448 ページの『HATS 管理コンソールの機能の使用』を参照してください。これらの各計画をカスタマイズするには、接続設定を追加または変更します。計画については、表 25 を参照してください。また、設定については、569 ページの表 26 を参照してください。

表 25. 使用可能な計画

計画	Java クラス名	説明と使用法
タイミング	TimingNextScreenBean	初期値は、待機時間として使用されます。表示スペースのすべてのデータを確実に受信するために、ホストのアウトバウンド・データ伝送の着信に基づいて、待機インターバルが追加されることがあります。追加されるインターバルは、初期値の半分の値に設定されます。HATS は、待機インターバルの後半に表示スペースのデータを受信すると、さらにインターバルを追加します。 この計画は、ホスト・タイプが TN5250、TN3270 の場合、およびホスト・タイプが TN3270E でコンテンション解消が使用されない場合に使用されます。
高速 3270E	Fast3270ENSB	この計画では、コンテンション解消で提供される TN3270E プロトコルによる通信を向上させ、アウトバウンド・データが完全に着信したタイミングをより正確に判断できます。この計画では、アウトバウンド・データ着信の追加タイミングはありません。そのため、 569 ページの表 26 で説明する **default.delayInterval** および **default.appletDelayInterval** 設定は無視されます。 この計画は、ホスト・タイプが TN3270E で、接続にコンテンション解消機能が使用される場合に、使用されます。コンテンション解消を HATS で使用するには、この機能を telnet サーバーで活動化し、接続開始時に telnet プロトコルで折衝する必要があります。詳細については、 573 ページの『コンテンション解消 (TN3270E のみ)』を参照してください。 この計画は、ホスト・セッションの SNA サイドがシステム・サービス制御点 (SSCP) に現在接続されている場合、ホスト・タイプ TN3270 と TN3270E にも使用されます。これは、HATS が TN3270E プロトコルを使用してアウトバウンド・データの着信をすぐに判断できるためです。
高速 5250	Fast5250NSB	この計画は、タイミング計画に基づいていますが、可能な場合、通常の待ちを簡略化することができます。OIA システム・ロック・フラグがクリアされ、空でない表示スペースの更新 (ブランク以外の文字を含む更新) をホストから受信している場合は、簡略化が許可され、通常の待ちが終了します。 この計画は、ホスト・タイプ TN5250 のデフォルトです。

画面間の遷移に関する設定のカスタマイズについては、表 26 を参照してください。

表 26. 画面間遷移設定

設定 (大文字と小文字を区別)	Description	デフォルト
default.delayInterval	画面確定計画を使用したアウトバウンド・データの分析中に、HATS が待機する通常の期間 (ミリ秒単位)。インターバルを追加するタイミング計画や簡略化を行う高速 5250 計画のように、画面確定計画によって、実際の期間がこの設定値と異なることがあります。より高い値を指定すると、長い処理時間やネットワーク遅延によりホストで遅延が発生した場合に、画面変換の信頼性が向上します。ただし、より高い値を設定すると、応答時間が長くなります。特に、タイミング計画を使用して、この値が画面間の最小遅延である場合に、この傾向が強くなります。 **注:** Fast3270E 計画では、この設定が使用されません。	1200 ミリ秒
default.appletDelayInterval	画面確定計画を使用したアウトバウンド・データの分析中に、HATS が待機する通常の期間 (ミリ秒単位)。非同期更新アプレットが使用可能であり、クライアント用に実行されている場合、default.delayInterval の代わりに、画面確定計画を使用します。インターバルを追加するタイミング計画や簡略化を行う高速 5250 計画のように、画面確定計画によって、実際の待機期間が異なることがあります。詳細については、576 ページの『自動最新表示』を参照してください。 **注:** Fast3270E 計画では、この設定が使用されません。	400
fast3270E.minimumWait	高速 3270E 計画と折衝されたコンテンション解消を使用したアウトバウンド・データの分析中に、HATS が待機する最小期間 (ミリ秒単位)。この設定のデフォルト値は、250 ミリ秒です。この設定は、最小待機期間の値が小さいほど、画面確定の信頼性が向上する、以下の 2 つの場合に便利です。 1. 高速 3270E 計画で、telnet クライアントで関連するすべてのイベントが処理される前に、受信したアウトバウンド・データのタイミングに応じて画面確定を実行できる場合 2. 変換を意図しない短期間の一時画面がホスト・アプリケーションから送信されるために、この設定を使用して、一時画面が意図された画面に更新されるまで HATS が待機する必要がある場合 この設定で指定される最小待機時間が全ての画面確定に使用されるので、この値を変更する場合は、ユーザー応答時間に注意してください。	250

表 26. 画面間遷移設定 (続き)

設定 (大文字と小文字を区別)	Description	デフォルト
fast5250.minimumWait	高速 5250 計画を使用したアウトバウンド・データの分析中に、HATS が待機する最小期間 (ミリ秒単位)。このオプション設定は、最小待機期間の値が小さいほど、画面確定の信頼性が向上する、以下の一般的な 2 つの場合に便利です。 1. 高速 5250 計画で、telnet クライアントで関連するすべてのイベントが処理される前に、受信したアウトバウンド・データのタイミングに応じて画面確定を実行できる場合 2. 変換を意図しない短期間の一時画面がホスト・アプリケーションから送信されるために、この設定を使用して、一時画面が意図された画面に更新されるまで HATS が待機する必要がある場合 この設定で指定される最小待機時間が全ての画面確定に使用されるので、この値を変更する場合は、ユーザー応答時間に注意してください。	0 (最小期間なし)
default.nonHostKeyWait	実際の画面確定ステップを必要としない [fldext]、[field+]、[eraseeof] などの擬似 AID キーの 1 つを処理した後に HATS が待機する期間をミリ秒単位で指定します。この設定を変更しないことをお勧めします。	100
nextScreenClass	アウトバウンド・データの分析中に計画のオーバーライドを許可します。それには、HATS が使用する計画とは別の計画の完全修飾 Java クラス名を指定します。ホスト・アプリケーションから 5250 アウトバウンド・データが送信され、HATS が一部のホスト画面または誤ったホスト画面を変換して表示する場合に、信頼性を向上させるために、5250 アプリケーションの調整でこの設定が使用されることがあります。このような特殊な場合は、この設定で **com.ibm.hats.runtime.TimingNextScreenBean** を指定することによって、Fast5250 計画でなく、タイミング計画がアウトバウンド・データ分析で使用されるように強制できます。HATS の計画に対する Java パッケージ名は、**com.ibm.hats.runtime** です。ホスト・タイプ TN3270 または TN3270E では、この設定を使用しないことをお勧めします。	ホスト・タイプなどの 568 ページの表 25 で説明する要因によって異なります。

すべての計画に適用される初期画面のカスタマイズ設定については、表 27 を参照してください。

表 27. 初期画面設定

設定 (大文字と小文字を区別)	説明	デフォルト
default.delayStart	アウトバウンド・データの分析中に、HATS が初期ホスト画面についてだけ待機する通常の期間 (ミリ秒単位)。接続セットアップ・プロセスが使用可能な画面を取得するためにすでに待機しているので、この値は、追加遅延にはなりません。578 ページの『関連する HATS 設定』の connecttimeout を参照してください。この値は、そのホストとの通信が開始されてから初期画面の認識を試行するまでに HATS が待機する最小期間です。高速ネットワークに対しては、この値を小さく設定し、初期画面の応答時間を向上させることができます。	2000

表 27. 初期画面設定 (続き)

設定 (大文字と小文字を区別)	説明	デフォルト
ignoreBlankStartupPS	ホストへの接続後に、完全にブランクである表示スペースを正しい最初の画面として使用するかどうかを指定します。ほとんどのホスト接続では、**MSG10** 画面などのログオン画面が表示されます。完全にブランクである画面も HATS で変換などの処理を行う有効な最初の画面として許容する場合は、この値を **false** に設定します。false を設定しない場合は、HATS で接続セットアップが成功したとは見なされません。	true

OIA フラグの取得待ち

HATS は、アウトバンド・データの分析後に、ホストの準備が整い、ユーザーがインバウンド・データを送信できる状態であることを示す OIA フラグを待ちます。アウトバウンド・データの分析後に、OIA フラグがすでにこの状態を示している場合もあります。その場合は、OIA フラグの待機はすぐに終了します。デフォルトでは、HATS は、STATE_SYS_LOCK (システム・ロック) フラグと STATE_TIME (キーボード入力禁止) フラグがクリアされるまで、最大 5 分間待機します。どちらの条件も、表 28 で定義する設定でカスタマイズできます。これらの設定は、アウトバウンド・データ分析に使用する計画に関係なく、適用されます。

表 28. OIA カスタマイズ設定

設定 (大文字と小文字を区別)	Description	デフォルト
keyboardInhibitedWait	キーボード入力禁止 OIA フラグがクリアされるまで待機するかどうかを指定します。この値を `false` に設定すると、HATS は、システム・ロック OIA フラグのクリアだけを待ちます。この設定は、キーボード入力禁止画面をクライアントに送信するために使用します。	true
oiaLockMaxWait	ユーザーが他のデータを送信できることを表す OIA フラグを HATS が待つ最大期間をミリ秒単位で指定します。ロック画面をより高速で戻す場合は、この値を小さくすることができます。この値が小さすぎると、画面の信頼性が低下します。特定のロック画面をユーザーに送信する場合は、oiaLockMaxWait の値を変更する前に keyboardInhibitedWait 設定を使用することをお勧めします。	300000 ミリ秒 (5 分に相当)。ただし、非同期更新アプレットが使用可能で実行されている場合を除きます。その場合のデフォルトは、 569 ページの表 26 で説明する default.appletDelayInterval のデフォルト値です。

カスタマイズ設定の変更

画面確定カスタマイズ設定は、HATS Toolkit の接続エディターを使用して変更できます。設定を変更するには、通常、「main」という名前の接続用の main.hco ファイルで「**HATS Projects**」ビューを開き、「**接続**」を展開して、「**main (デフォルト)**」を選択します。

以下の 5 つの設定は、接続エディターの「**画面処理 (Screen Handling)**」タブで変更できます。

- default.delayInterval
- default.delayStart
- default.appletDelayInterval
- default.blankScreen

  577 ページの表 29 を参照してください。

- default.blankScreenKeys

  577 ページの表 29 を参照してください。

前述のすべての設定は、「**ソース**」タブで構成できます。<classSettings> タグ内では、画面確定設定が **com.ibm.hats.common.NextScreenSettings** クラス内に含まれます。

画面確定パフォーマンスの必要に応じて、設定を追加または変更できます。デフォルトの接続設定は、以下のとおりです。

```
<classSettings>
    ...
   <class name="com.ibm.hats.common.NextScreenSettings">
      <setting name="default.delayInterval" value="1200"/>
      <setting name="default.appletDelayInterval" value="400"/>
      <setting name="default.blankScreen" value="timeout"/>
      <setting name="default.delayStart" value="2000"/>
   </class>
</classSettings>
```

以下の例では、1 つの設定が更新され、1 つの設定が追加されます。

```
<classSettings>
...
   <class name="com.ibm.hats.common.NextScreenSettings">
      <setting name="default.delayInterval" value="1200"/>
      <setting name="default.appletDelayInterval" value="400"/>
      <setting name="default.blankScreen" value="timeout"/>
      <setting name="default.delayStart" value="1500"/>
      <setting name="nextScreenClass"
               value="com.ibm.hats.runtime.TimingNextScreenBean"/>
   </class>
</classSettings>
```

HATS で使用されている計画の判断

HATS で使用されている計画を判断するには、以下のステップを実行します。

1. HOD 転送と HATS ランタイムのトレースをオンにします。トレースの詳細については、 448 ページの『HATS 管理コンソールの機能の使用』を参照してください。runtime.properties ファイル内の設定は、以下のとおりです。

   ```
   trace.HOD.TRANSPORT=2
   trace.RUNTIME=7
   ```

2. トレース・ファイル (通常は、HATS.ear ファイルの **logs** ディレクトリー内に格納されており、**trace** で始まるファイル名を持つ) を表示します。
3. HATS による画面確定時の以下のような項目を検索します。

```
Text        RUNTIME      runtime.NextScreenBean.getInstance()
   oiaStatusFlags/isSSCP=x7 false

 Text        RUNTIME      runtime.Fast3270ENSB.<init>()
   Create: Fast3270E: initialConnect=false, ttwOriginal=1200, tStartedWaiting=0,
                      tLastPSEvent=0, keyToSend=[enter], minimumWait=250
```

太字で強調表示されたキー・テキストは、画面確定処理を開始する最初の項目例のメソッド名 **runtime.NextScreenBean.getInstance()** です。この項目で作成されたオブジェクトの名前は、`Create:` で始まります。この例で作成された計画オブジェクトの名前 (Fast3270E) とその Java クラス名 (Fast3270ENSB) は、2 つ目のトレース項目例で*斜体* で強調表示されています。

コンテンション解消 (TN3270E のみ)

コンテンション解消は、Internet Engineering Task Force (IETF) の TN3270E プロトコル規格 (RFC2355) の最新の草案で定義され、多数の TN3270E サーバーで使用可能です。この機能は、ホストの SNA プロトコルからクライアントの Telnet プロトコルへの変換に関する制限を克服するのに役立ちます。コンテンション解消を使用すると、HATS を含む TN3270E クライアントのパフォーマンスが向上します。

コンテンション解消の使用は、HATS の telnet クライアント (HACL telnet クライアント) と TN3270E サーバーの間で、接続セットアップ中に折衝されます。HATS と TN3270E サーバーの間でコンテンション解消の折衝に成功した場合、より効率的な通信を実現でき、ホスト・システムとの対話の遅延が減少します。デフォルトでは、HATS は、TN3270E ホスト・タイプの 3270 ベースのホスト・システムとの間でコンテンション解消の折衝を試行します。

z/OS Communications Server を使用する場合のコンテンション解消

z/OS 1.2 以降、IBM は、VTAM を含め、z/OS Communications Server におけるコンテンション解消をインプリメントしてきました。ただし、z/OS 1.2、1.3、および 1.4 のオリジナルのインプリメンテーションは、急拡大する規格に完全には一致していませんでした。その結果、 HATS がこれらのオリジナル・リリースの z/OS Communications Server と通信しにくいことがありました。この問題を解決するために、IBM は、パッチ (PTF) を発行し、推奨されるパッチをリストした情報 APAR (テクニカル・ノート) を公開していました。

z Systems 管理者に連絡して、APAR II13135「Common Telnet Problems Under z/OS」を参照してください。APAR II13135 では、HATS でコンテンション解消を活用するために推奨される PTF がリストされています。例えば、z/OS 1.4 の場合、APAR PQ71574 と PQ72265 が適用され、PTF UQ76065 で修正されます。正しいフィックスがない場合、HATS がハングすることがあります。この問題が発生すると、HATS Toolkit の端末ウィンドウにクロック画面 (クロック・アイコンが下部に表示されたまま画面がフリーズした状態) が表示されたままになるので、HATS 開発者はこの問題の発生を確認できます。

z Systems 管理者が II13135 パッチをすぐにインストールできない場合は、 578 ページの『関連する HATS 設定』で説明する **negotiateCResolution** 設定を使用して、値 `false` を指定します。この設定に `false` を指定すると、HATS がコンテンショ

ン解消の折衝をスキップし、以前の TN3270E の仕様に準拠します。コンテンション解消を使用不可にすることにより、3270 の応答時間が長くなり、HATS のパフォーマンスが低下することがあります。使用不可にしなければ悪影響はありません。

コンテンション解消の折衝に関するこの問題は、HATS が、影響がある z/OS ホスト・システムに直接接続せずに、フロントエンドの 3270 通信ゲートウェイ経由で接続する場合には、発生しません。これらのゲートウェイでは、コンテンション解消がサポートされない場合、またはコンテンション解消をサポートするためにサービスを追加する必要がある場合があります。

他の Communications Server を使用する場合のコンテンション解消

AIX、Linux、Linux on zSeries、および Windows 向けの IBM Communications Server の最新リリースでも、コンテンション解消が正しくサポートされます。パッチによる z/OS Communications Server 1.2 以降へのアップグレードが不可能な場合は、これらの代わりのプラットフォーム・バージョンの Communications Server を使用して、HATS でコンテンション解消による応答の高速化を実現できます。IBM Communications Server も、VM、VSE、または TPF を実行中で、コンテンション解消がサポートされないホスト・システムに対して役立つことがあります。

これらのインスタンスでは、HATS は TN3270E 経由で、オプションで SSL や TLS を使用して、AIX、Linux、Linux on zSeries、または Windows で実行中の IBM Communications Server に接続します。IBM Communications Server は、SNA 接続 (従来の SNA 接続、またはエンタープライズ・エクステンダー (インターネット・プロトコル・ネットワークを経由する SNA トラフィック)) を使用してホスト・システムに接続します。システム管理者は、ホスト・システムではなく、IBM Communications Server で TN3270E LU を管理できます。

コンテンション解消を使用する場合のパフォーマンスへの影響

HATS のデフォルト設定を使用すると、コンテンション解消によって 3270 画面間の伝送遅延を約 800 ミリ秒短縮することができます (IBM による制約がない高速ネットワークでのテスト結果)。通常の Web ユーザーにとってこの程度の遅延は許容範囲内にありますが、コンテンション解消を使用すると、特定の高速度ユーザー、およびハイパフォーマンスの Web サービスの場合にメリットがあります。

コンテンション解消は、特定のワークロードを処理するための HATS サーバー・プロセッサーの処理能力要件に直接影響しません。

コンテンション解消状況の判断

HATS が特定の TN3270E 接続について正しくコンテンション解消を折衝しているかどうかを判断できます。HATS のトレースを調べて折衝の結果を表示するには、以下のステップを実行します。

1. HOD 転送と HATS ランタイムのトレースをオンにします。トレースの詳細については、 448 ページの『HATS 管理コンソールの機能の使用』を参照してください。runtime.properties ファイル内の設定は、以下のとおりです。

```
trace.HOD.TRANSPORT=2
trace.RUNTIME=7
```

2. トレース・ファイル (通常は、HATS.ear ファイルの **logs** ディレクトリー内に格納されており、**trace** で始まるファイル名を持つ) を表示します。
3. トレース・ファイル内で、接続開始時の以下のような項目をまとめて検索します。

```
Text        HOD          HOD.()
   Correlator: HodConn:localhost:localhost:server1:HATS.ear#1
   trcmsg: -->TN3270 : Negotiate     CMD = DO      OPT = TN3270-E

 Text        HOD          HOD.()
   Correlator: HodConn:localhost:localhost:server1:HATS.ear#1
   trcmsg: <--TN3270 : Response      CMD = WILL    OPT = TN3270-E
```

   キー・テキストは、**太字**で強調表示された、CMD = DO と CMD = WILL です。OPT = TN3270-E (ここでは太字で表示しています) も同時に現れます。CMD = DO と CMD = WILL の両方が表示される場合、HATS が特定のホスト接続について TN3270E と折衝しており、コンテンション解消を要求できることを表します。CMD = WONT と CMD = DONT の一方または両方が表示される場合は、ホストで TN3270E がサポートされず、HATS がコンテンション解消を要求できない可能性があります。
4. 接続開始時の以下のような項目を検索してください。

```
Text        HOD          HOD.()
   Correlator: HodConn:localhost:localhost:server1:HATS.ear#1
   trcmsg: <--TN3270 : FUNCTION_STATUS: CR REQUESTED
```

   キー・テキストは、**太字**で強調表示された、**FUNCTION_STATUS: CR REQUESTED** です。このテキストが表示された場合は、HATS が特定のホスト接続についてコンテンション解消を要求していることを示します。
5. 接続開始時の以下のような項目を検索してください。

```
Text        HOD          HOD.()
   Correlator: HodConn:localhost:localhost:server1:HATS.ear#1
   trcmsg: <--TN3270E: FUNCTION_STATUS: CR ENABLED
```

   キー・テキストは、**太字**で強調表示された、**FUNCTION_STATUS: CR ENABLED** です。このテキストが表示された場合は、HATS が特定のホスト接続に関するコンテンション解消の折衝に成功したことを示します。
6. HATS による画面確定時の以下のような項目を 2 つ検索します。

```
Text        RUNTIME      runtime.NextScreenBean.getInstance()
   isEnhancedTNServer=true

 Text        RUNTIME      runtime.Fast3270ENSB.<init>()
   Create: Fast3270E: initialConnect=false, ttwOriginal=1200,
                                            tStartedWaiting=0,tLastPSEvent=0,
                                            keyToSend=[enter], minimumWait=250
```

   キー・テキストは、**太字**で強調表示された、**EnhancedTNServer=true** と **Create: Fast3270E** です。これらが表示された場合、HATS がコンテンション解消の折衝に成功し、アウトバウンド・データ分析のために Fast3270E 計画を使用して応答していることを示します。

自動最新表示

HATS は、クライアント・ブラウザーを非同期アウトバウンド・データで更新するために、クライアント・プル機能とサーバー・プッシュ機能の両方を提供します。これらの機能の詳細については、 134 ページの『自動切断および最新表示 Web のみ』を参照してください。

これらの機能を使用して、HATS の画面確定処理を調整することもできます。これらの機能を使用するかどうかは、以下に基づきます。

- 必要な場合、アプリケーションが画面確定設定の微調整をどのように処理するか。
- TN3270E ホストのコンテンション解消がオプションであるかどうか。
- ポータルやファイアウォールなどの考慮事項がデプロイメントでの使用を制限するかどうか。

これらの機能のいずれかを使用すると、画面確定回数を減らし、パフォーマンスを向上させることができます。これは、失敗した更新をアプリケーション・サーバーで受信したときに、これらの更新をクライアントで最新表示できるためです。一部のデフォルト画面確定設定は、クライアントでサーバー・プッシュ (アプレット) 機能を構成および実行するときに、変更されます。これらの設定について詳しくは、569 ページの表 26 および 571 ページの表 28 を参照してください。

一時画面の処理

一時画面は、ユーザーが応答するアプリケーション画面 (意図された画面) を表示するまでの最低限の期間だけクライアントで表示するために、ホストから意図的に送信されるホスト画面です。一時画面の 2 つの例は、以下のとおりです。

- 「お待ちください」というメッセージ、またはその他の処理中を表すメッセージ
- アプリケーション内、アプリケーション間、またはログオン画面から他の画面への移動中にホストが発行した画面クリア・コマンドによって生成されたブランク文字だけを含むブランク画面。

注: マクロでの一時画面の処理については、 396 ページの『一時画面の処理』を参照してください。

標準のエミュレーターを実行しているユーザーは、一時的に表示され、意図された画面に置換される一時画面に気付かないこともあります。HATS が一時画面を受信すると、ホストがデータ送信を完了したと HATS が誤って判断することがあります。一時画面は、使用可能な画面カスタマイズと比較され、ほとんどの場合、デフォルトの変換によってレンダリングされて、クライアント・ブラウザーに表示されます。一時画面と置換するための意図された画面がホストから送信されますが、HATS では、そのブラウザー更新に関する画面確定が完了しているため、ホスト画面更新を待つことも処理することもありません。意図されていない一時画面が変換されていることにユーザーが気付いて、**「最新表示」** をクリックすると、HATS は、現在のホスト画面と同期し、クライアント・ブラウザーが正しく更新されます。この結果は全体としては不本意であり、通常は画面確定までの遅延時間の調整により、場合によっては自動最新表示機能のいずれかを使用可能にすることにより防止できます。

望ましくないこのシナリオを除去する別の方法は、一時画面に一意に一致し、他の画面に一致しない 1 つ以上の画面カスタマイズを作成して、適切なアクションを取ることです。このアクションには何も組み込むことができません。このアクションは、短い <pause> エレメントのみを含む簡単なマクロの「マクロを実行」アクションを使用してインプリメントできます。画面カスタマイズ・アクション・リストの処理後に HATS が実行する画面確定アクションと組み合わせると、ブラウザー更新要求に応答するための目的の画面を受信するまで HATS が待機する結果になります。画面カスタマイズに認識基準を作成する場合は、ユーザーに対して表示する画面と一致しないように注意する必要があります。

HATS では、追加の画面確定設定が提供されます。これらの設定により、画面カスタマイズとマクロを作成しなくてよいため、ブランク画面の処理が簡単になります。ブランク画面の処理を簡略化するこれらのカスタマイズ設定では、HATS がブランク画面を確定した場合に実行するアクションを指定できます。指定できるアクションは、待機、特定のホスト・キーの送信、画面認識処理への移行のいずれかです。

初期画面でのブランク画面処理については、2 つの追加画面確定接続設定を使用できます。初期ブランク画面処理の構成については、157 ページの『第 6 章 接続の管理』の 165 ページの『画面処理』を参照してください。

初期ブランク画面処理設定

初期ブランク画面処理設定については、表 29 を参照してください。

表 29. ブランク画面処理設定

設定	Description	値	デフォルト
default.blankScreen	システムは、「拡張」タブの接続タイムアウトで指定された時間、非ブランク画面を待ちます。default.blankScreen 設定により、ブランク画面のみの状態が続いて接続タイムアウトの有効期限が切れた場合の処理を指定します。有効な値は以下のとおりです。 **timeout** エラー・メッセージを発行します。これはデフォルト値です。 **normal** ブランク画面が表示されます。初期画面がブランク画面になることが予期される場合は、低めの接続タイムアウトを指定することもできます。 **sendkeys** default.blankScreen.keys 設定で定義されたホスト・キーが送信されます。初期画面がブランク画面になることが予期される場合は、低めの接続タイムアウトを指定することもできます。	タイムアウト、通常、または sendkeys	タイムアウト
default.blankScreen.keys	`default.blankScreen` が `sendkeys` に設定されている場合に送信するホスト・キーを指定します。	有効ホスト・キー	なし

初期画面より後に表示されるブランク画面には、定義されたアプリケーション・イベントが使用されます。このアプリケーション・イベントについては、173 ページの

の『第 7 章 画面イベントの処理』の 125 ページの『アプリケーション・イベント』にある「ブランク画面」を参照してください。

関連する HATS 設定

以下の設定では、HATS 画面確定を直接構成できません。ただし、画面確定に影響するため、参考のためにここで説明します。

名前	設定場所	説明
negotiateCResolution	この接続プロパティーは、HATS Toolkit の接続エディターにある「**拡張**」タブの拡張接続設定テーブルに追加することができます。 .hco ファイルのソースでは、このプロパティーは <otherParameters> タグの項目になります。	HATS で使用される telnet クライアント (HACL) に、TN3270E 接続でコンテンション解消を折衝するかどうかを通知します。デフォルトは `true` です。 このパラメーターを `false` に設定すると、HATS は telnet サーバーからのコンテンション解消を要求しません。コンテンション解消は使用されないため、このホストの HATS 画面確定動作が実行され、タイミング計画が使用されます。 この機能とこの機能を使用不可に設定する理由について詳しくは、 573 ページの『コンテンション解消 (TN3270E のみ)』を参照してください。 コンテンション解消は、ホスト・タイプが TN3270E の場合だけに要求されるので、ホスト・タイプを TN3270E ではなく TN3270 に変更すると、副次作用として HATS 画面確定が同じ影響を受けます。

名前	設定場所	説明
connecttimeout	HATS 接続エディターの「**基本**」では、この接続プロパティーに「**指定された秒数の経過後に接続試行を中止 (Abandon attempt to connect after specified number of seconds)**」というラベルが付けられます。 .hco ファイルのソースでは、これは <hodconnection> タグの connecttimeout 属性です。 **注:** これは、接続エディターで拡張接続プロパティーを追加するときにドロップダウン・リストに表示される connectionTimeout パラメーターでは**ありません**。これは、新規のプロパティーであり、HATS が使用可能なプロパティーのリストを要求したときに、Host On-Demand 製品から HATS に公開されます。HATS パラメーターまたは HATS 画面確定とは関係ありません。	接続プール (プールが使用可能な場合) または新規接続から、接続が使用可能になるまでサーバーが待機する時間を秒単位で指定します。デフォルトは、120 ミリ秒です。新規接続は、ホストとの通信が開始され、使用可能なホスト画面を受信したときに、使用可能になります。完全にブランクの画面は、ignoreBlankStartupPS 画面確定フラグが `false` に設定されない限り、使用不可と見なされます。

名前	設定場所	説明
• ignorepauseforenhancedtn • ignorepauseoverrideforenhancedtn • delayifnotenhancedtn	これらのマクロ・プロパティーは、マクロのソース・ビューで指定できます。詳しくは、「*拡張マクロ・ガイド*」の『タイミングの問題』を参照してください。	ホストによるデータ送信の完了時には、マクロでも画面確定と同じ状況が発生します。そのため、「*拡張マクロ・ガイド*」で説明するこれらのプロパティーおよびその説明をここでも紹介します。 これらのマクロのプロパティーでは、コンテンション解消を使用できる環境と使用できない環境の両方でマクロ開発者が単一バージョンのマクロをサポートしようとする際に直面する問題を解決できます。これらのプロパティーを使用して、コンテンション解消を使用できない環境に必須であるマクロの <pause> エレメントを、コンテンション解消を使用する環境でどのように実行するかを制御します。 <HAScript> のプロパティー **ignorepauseforenhancedtn** のデフォルト値は、`false` です。この値は、<pause> エレメントが常に処理されることを表します。この値を `true` に変更すると、コンテンション解消の使用時に <pause> エレメントが無視されます。 <pause> のプロパティー **ignorepauseoverrideforenhancedtn** を `true` に設定すると、<pause> エレメントがスキップされずに、必ず実行されます。<HAScript> エレメントの **ignorepauseforenhancedtn** を `true` に設定した、コンテンション解消を使用可能な環境でも、同様に <pause> エレメントが実行されます。 **ignorepauseforenhancedtn** の値が `true` の場合は、コンテンション解消を活用するために、<HAScript>のプロパティー **delayifnotenhancedtn** を必ず使用する必要があります。マクロは、コンテンション解消を使用できない環境でも使用されることがあります。このプロパティーでは、OIA の変更時に発生する遅延をミリ秒単位で指定します。デフォルトは、0 です。 これらのプロパティーについて詳しくは、「*拡張マクロ・ガイド*」の『タイミングの問題』を参照してください。

HATS 画面確定の調整

HATS 画面確定の信頼性とパフォーマンスの両方を向上させるために、HATS 画面確定をさまざまな方法で調整できます。その場合、以下のような事項を決定する必要があります。

パフォーマンスと信頼性の向上に、コンテンション解消を使用するかどうか。

HATS でコンテンション解消の折衝に成功した場合、より効率的な通信を実現でき、ホスト・システムとの対話の遅延が減少します。詳細については、573 ページの『コンテンション解消 (TN3270E のみ)』を参照してください。

計画の遅延を調整するかどうか。

default.delayInterval を変更して、応答時間が長いアプリケーションや環境での応答時間の向上や、信頼できるパフォーマンスの確保などを実現する必要がある場合があります。ホスト・システムによっては、特定の画面を送信するまでに長い応答時間の遅延が繰り返し発生することがあります。遅延は、大量の処理が必要なホストのステップで発生することがあります。また、ときどき要求が多くなるネットワーク・リソースが遅延に関係することもあります。これらのケースでは、delayInterval の値を大きくすると、画面確定の信頼性が向上することがあります。ホストの平均応答時間だけでなく、典型的な最長変動も考慮する必要があります。典型的な操作条件に適した値を選択してください。同様に、default.delayStart、default.appletDelayInterval、fast3270E.minimumWait、および fast5250.minimumWait 設定でも、さまざまな場合の遅延を制御できます。569 ページの表 26 で設定の説明を参照してください。

default.delayInterval のような遅延値の設定が小さすぎる場合にどうなるか。

HATS ランタイムで、画面の完成とホストからのデータ送信の完了を判断できます。これらの徴候の例は、以下のとおりです。

- HATS 画面がブランクの場合に「最新表示」をクリックすると、正しい画面が表示される。
- 画面が部分的に (不完全に) レンダリングされているように見える場合に、「最新表示」をクリックすると、正しい画面が表示される。
- 誤った画面カスタマイズ・イベントが呼び出される。この場合、画面認識基準で不完全な画面を正しく検査できないため、誤ったカスタム変換が表示されることがある。

遅延値の設定が大きすぎる場合にどうなるか。

ユーザーに対する応答時間が長くなり、最悪の場合は、画面を取得するための待機が終了します。アプリケーション・チューニングでは、アプリケーションとそのアプリケーションが使用されるネットワーク環境について運用面の判断を行い、バランスを保つ必要があります。

他の問題があるシナリオを修正できるかどうか。

計画の遅延を調整することによって解決される状況に似ていても、ソリューションが異なる場合があります。

- アプリケーションで使用禁止画面を処理できるようにします。この場合、OIA フラグの取得待機に関するカスタマイズ設定 **keyboardInhibitedWait** と **oiaLockMaxWait** を使用します。
- 5250 接続の HATS での画面更新は、計画の遅延を調整した後でも失敗することがあります。**nextScreenClass** 設定を使用して、Fast5250 計画をオフにする必要が生じることがあります。

自動最新表示を使用するかどうか。

ご使用のネットワーク環境と要件に基づいて、自動最新表示を使用すると応答時間の向上に役立つことがあります。詳細については、576 ページの『自動最新表示』を参照してください。

リファレンス

この節のリファレンスは、画面確定の調整に便利です。

コンテンション解消の情報

RFC2355 拡張機能 (2003 年 3 月ドラフト)

http://www.ietf.org/proceedings/03mar/I-D/draft-ietf-tn3270e-extensions-04.txt

情報 APAR II13135

http://www.ibm.com/support/docview.wss?uid=isg1II13135

z/OS Communications Server

z/OS Knowledge Center (http://www-01.ibm.com/support/knowledgecenter/SSLTBW_1.12.0/com.ibm.zos.r12/zosr12home.html) にアクセスして用語「SNAEXT」で検索してください。

Linux、Linux on zSeries、AIX、および Windows 向けの IBM Communications Server http://www.ibm.com/software/network/commserver

拡張マクロ・ガイド

マクロ設計のパラメーターおよびオプションの影響を受けるマクロ再生中のタイミングの問題、およびマクロに関する画面確定の問題については、「*拡張マクロ・ガイド*」の『タイミングの問題』を参照してください。

付録 C. 特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。 本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。 IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

© Copyright IBM Corp. 2003, 2015

Intellectual Property Dept. for Rational Software
IBM Corporation
5 Technology Park Drive
Westford, MA 01886
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、 IBM より提供されます。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。 IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態で提供されるものであり、いかなる保証も提供されません。IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

この情報をソフトコピーでご覧になっている場合は、写真やカラーの図表は表示されない場合があります。

プログラミング・インターフェース情報

このユーザーと管理者のガイドには、プログラムを作成するユーザーが HATS のサービスを使用するためのプログラミング・インターフェースに関する情報があります。

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、http://www.ibm.com/legal/copytrade.shtml をご覧ください。

Adobe は Adobe Systems Incorporated の米国およびその他の国における登録商標です。

Linux は、Linus Torvalds の米国およびその他の国における商標です。

Microsoft および Windows は、Microsoft Corporation の米国およびその他の国における商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

用語集

アクション (action). 画面イベントに指定された画面認識基準と一致するホスト画面などの、イベントの結果として管理対象オブジェクトに対してアプリケーションが実行する既定のタスク。アクションのリストは、各イベントの定義の一部である。

ADB. 「**アプリケーション・データ・バッファー (application data buffer)**」を参照。

管理コンソール (administrative console). HATS 管理コンソールは、Web ベースのユーティリティーである。管理コンソールで提供されるビューおよび機能を使用すると、HATS Web アプリケーションのためのライセンスと接続の管理、ログ設定とトレース設定、メッセージとトレースの表示、および問題判別を実行できる。

アプリケーション (application). 「**HATS アプリケーション (HATS application)**」を参照。

アプリケーション・データ・バッファー (application data buffer). WebFacing アプリケーションが消費するために WebFacing サーバーによって戻されるデータの形式。

アプリケーション・イベント (application event). アプリケーションのライフ・サイクル内での状態変更により起動される HATS イベント。アプリケーション・イベントの例には、ユーザーによる HATS アプリケーションへの最初のアクセス (開始イベント)、またはアプリケーションによる認識できない画面の検出 (一致しない画面イベント) などがある。

アプリケーション・キーパッド (application keypad). HATS アプリケーション・レベルの機能を表す一連のボタンまたはリンク。 (「**ホスト・キーパッド (host keypad)**」と対比。)

成果物 (artifact). 「**リソース (resource)**」を参照。

バックグラウンド接続 (background connection). デフォルト接続以外の、HATS アプリケーションで定義されている接続。 HATS はバックグラウンド接続からの画面は変換しない。 (「**デフォルト接続 (default connection)**」と対比。)

双方向 (bidi) (bidirectional (bidi)). 一般に右から左に読み、数字だけは左から右に読むアラビア語やヘブライ語などのスクリプトに関係した表現。

BMS マップ (BMS map). CICS の基本マッピング・サポートで使用する画面定義ファイル。 BMS マップは、CICS アプリケーションによりグループとして表示される一連のフィールドを定義する。

ビジネス・ロジック (business logic). 高度な機能を実行する Java コード。例えば、Java API 経由でアクセス可能な別のアプリケーション、データベース、またはその他のシステムと対話する機能を実行する。ビジネス・ロジックは、アプリケーションまたは画面イベント内のアクションとして呼び出される。

チェックイン画面 (checkin screen). ホスト画面を識別する画面。これは接続プールに戻せると見なされる接続に対してアクティブである必要がある。アプリケーションがチェックイン画面で指定された画面上にない場合、接続は廃棄されるか、またはチェックイン画面で指定されたホスト画面への接続を戻そうとしてリサイクルされる。チェックイン画面は、接続に対して接続プールが指定されている場合にのみ意味がある。

コンポーネント (component). ホスト画面のビジュアル要素。コマンド行、ファンクション・キー、または選択リストなど。 HATS アプリケーションは、ホスト・コンポーネントをウィジェットに変換する。

接続 (connection). HATS により使用される一連のパラメーター。.hco ファイルに保管され、ホスト・アプリケーションに接続する。 (「**デフォルト接続 (default connection)**」および「**バックグラウンド接続 (background connection)**」も参照。)

接続プール (connection pool). 初期状態で維持されるホスト接続のグループ。作成および初期化せずに使用できる。

© Copyright IBM Corp. 2003, 2015

信任状マッパー (credential mapper). Web 高速ログオンのコンポーネント。ネットワーク・セキュリティー層により事前に認証されている、ホスト信任状に対する要求を処理する。 (「**ネットワーク・セキュリティー層 (network security layer)**」を参照。)

DDS マップ (DDS map). データ記述仕様 (Data Description Specification) マップの略。これらのマップでは、IBM i 端末アプリケーションの表示スペースのレイアウトおよび動作を定義する。

デバッグ (Debug). リッチ・クライアント・プロジェクトの場合、「実行」と同じである。加えて以下の操作を実行できる。

- 端末の表示を使用して、プロジェクトのテスト中にホスト画面ナビゲーションを表示する。
- Rational SDP コンソールにデバッグ・メッセージを表示する。
- アプリケーションを再始動せずに、プロジェクトへの変更 (テンプレートまたは変換の変更など) を表示する。
- ランタイム環境に展開された設定 (runtime.properties ファイル内で定義) を変更せずに、ランタイム設定 (runtime-debug.properties ファイル内で定義) を変更およびテストする。
- HATS ビジネス・ロジックなどの Java コードをステップスルーする。

サーバーでデバッグ (Debug on Server). Web プロジェクトの場合、「サーバーで実行」と同じである。加えて以下の操作を実行できる。

- 端末の表示を使用して、プロジェクトのテスト中にホスト画面ナビゲーションを表示する。
- Rational SDP コンソールにデバッグ・メッセージを表示する。
- テスト・サーバー上でアプリケーションを再始動せずに、プロジェクトへの変更 (テンプレートまたは変換の変更など) を表示する。
- ランタイム環境に展開された設定 (runtime.properties ファイル内で定義) を変更せずに、ランタイム設定 (runtime-debug.properties ファイル内で定義) を変更およびテストする。
- HATS ビジネス・ロジックなどの Java コードをステップスルーする。

デフォルト接続 (default connection). HATS がホスト・アプリケーション画面を変換してユーザーに提示する接続。「**変換接続 (transformation connection)**」とも言う。 (「**バックグラウンド接続 (background connection)**」と対比。)

デフォルト・レンダリング (default rendering). 特定の変換が指定されていないホスト画面のパーツのレンダリングに HATS が使用するメソッド。

展開 (deploy). HATS アプリケーションをランタイム環境内で使用できるようにすること。 HATS Web アプリケーションの場合、これには、HATS プロジェクトの Java EE アプリケーション (つまり .ear ファイル) としてのエクスポート、および WebSphere Application Server へのそのインストールが含まれる。HATS リッチ・クライアント・アプリケーションの場合、これには、HATS プロジェクトの Eclipse フィーチャーとしてのエクスポート、および個々のクライアント・システムへのそのインストールが含まれる。インストールは、独立の Eclipse アプリケーションとして行う場合と、更新サイトから既存の Eclipse ランタイム環境に対して行う場合がある。

記述子 (descriptor). 「**画面認識基準 (screen recognition criteria)**」を参照。

開発者 (developer). HATS Toolkit を使用してアプリケーションを開発する人。アプリケーション開発者または Web 開発者とも呼ばれる。 (「**ユーザー (user)**」と対比。)

Device Runtime Environment (DRE). Lotus Expeditor Client V6.2.0 以前で HATS リッチ・クライアント・アプリケーションを実行するために必要な他のランタイム環境 (J2SE ランタイムなど) を含むパッケージ。DRE は、Lotus Expeditor Client のランタイム環境にインストールされる。

端末の表示 (display terminal). 実行時の HATS アプリケーションとホスト・アプリケーションとの対話を監視するために、テストおよびデバッグ中に使用できるホスト画面を表示する端末ウィンドウ。端末ウィンドウ内のホスト画面を使用して、ホスト・アプリケーションと対話することもできる。

Eclipse. ISV および他のツール開発者に、プラグ・コンパチブル・アプリケーション開発ツール用の標準プラットフォームを提供する、オープン・ソース・イニシアチブ。 Eclipse は、http://www.eclipse.org からダウンロードできる。

エディター (editor). 既存のデータを変更できるアプリケーション。 HATS Toolkit では、エディターはウィザードで作成されたリソースのカスタマイズに使用できる。

拡張非プログラマブル端末ユーザー・インターフェース (ENPTUI: Enhanced Non-Programmable Terminal User Interface). 非プログラマブル端末 (NPT) およびプログラマブル・ワークステーション (PWS) 上で拡張インターフェースを使用可能にし、5250 フルスクリーン・メニュー方式インターフェース上で、5250 表示データ・ストリーム拡張機能を使用する。

エンタープライズ・アーカイブ (EAR: enterprise archive). 特殊な Java アーカイブ (JAR)ファイル。Java EE 標準により定義され、Java EE アプリケーションを Java EE アプリケーション・サーバーに展開するために使用される。EAR ファイルには、エンタープライズ Bean、配置記述子、および個々の Web アプリケーション用の Web アーカイブ (WAR) ファイルが含まれる。 (Sun)

Enterprise JavaBeans (EJB). オブジェクト指向の分散エンタープライズ・レベル・アプリケーションの開発およびデプロイメントのために Oracle によって定義されたコンポーネント・アーキテクチャー。 (Oracle)

イベント (event). 特定の状態への到達に基づいて一連のアクションを実行する HATS リソース。アプリケーション・イベントと画面イベントの、2 つのタイプの HATS イベントがある。

エクスポート (export). アプリケーションの展開に備えて、HATS プロジェクトのリソースを収集すること。リソースは、必要な実行可能コードとともに、(Web アプリケーションの場合は) アプリケーション EAR ファイル、(リッチ・クライアント・アプリケーションの場合は) Eclipse フィーチャーに収集される。

Extensible Markup Language (XML). SGML から派生し、そのサブセットであるマークアップ言語を定義するための標準メタ言語。

GB18030. GB18030 は、新規の中国語文字エンコード規格である。 GB18030 には 160 万の有効なバイト・シーケンスがあり、1、2、または 4 バイトの文字シーケンスをエンコードする。

グローバル規則 (global rule). 特定の入力フィールドのレンダリングを特定の基準に基づいて変更する方法を定義する規則。グローバル規則は、カスタマイズ画面およびデフォルトを使ってレンダリングされる画面に使用される。グローバル規則は、プロジェクト・レベルまたは画面イベント・レベルで定義できる。

グローバル変数 (global variable). アクション用の情報を含めるために使用する変数。グローバル変数の値は、ホスト画面などから抽出して、テンプレート、変換、マクロ、統合オブジェクト、またはビジネス・ロジックで使用することができる。グローバル変数は、単一値にも配列にもでき、同じブラウザー・セッションを共用する他の HATS アプリケーションと共用できる。

HATS. 「**Host Access Transformation Services**」を参照。

HATS アプリケーション (HATS application). ホスト・アプリケーションの特定のバージョンをユーザーに提示するアプリケーション。WebSphere Application Server に展開された Web 対応アプリケーション、WebSphere Portal に展開されたポートレット、または Lotus Notes、Lotus Expeditor Client などの Eclipse リッチ・クライアント・プラットフォームに展開された Eclipse クライアント・サイド処理プラグインの形を取る。 HATS アプリケーションは、HATS プロジェクトから HATS Toolkit で作成され、適用可能な環境に展開される。展開されたアプリケーションは、他のホストまたは e-ビジネス・アプリケーションと対話して、組み合わせた情報をユーザーに提示できる。

HATS EJB プロジェクト (HATS EJB project). 他のアプリケーションがホスト・データを入手するために使用できる、HATS EJB および統合オブジェクトを含むプロジェクト。 HATS EJB プロジェクトは、ホスト・アプリケーションからの変換済み画面は示さない。

HATS エントリー・サーブレット (HATS entry servlet). HATS Web アプリケーションをブラウザーで開始するときに処理されるサーブレット。

HATS プロジェクト (HATS project). HATS リソースの集合 (成果物とも呼ばれる)。HATS Toolkit ウィザードを使用して作成され、HATS Toolkit エディターを使用してカスタマイズされ、HATS アプリケーションにエクスポートできる。

HATS Toolkit. Rational SDP 上で稼働する HATS のコンポーネント。これにより、HATS プロジェクトの作業を行って HATS アプリケーションを作成できるようになる。

Host Access Transformation Services (HATS). IBM ソフトウェアのツール・セットであり、ホスト・ベース・アプリケーションおよびデータ・ソースへの Web ベースのアクセスを可能にする。

ホスト・コンポーネント (host component). 「**コンポーネント (component)**」を参照。

ホスト・キーパッド (host keypad). ファンクション・キーまたは Enter キーなどの、一般にホスト・キーボードから使用できる機能を表す一連のボタンまたはリンク。 (「**アプリケーション・キーパッド (application keypad)**」と対比。)

ホスト・シミュレーション (host simulation). ホスト・シミュレーションを利用すると、ホスト・シミュレーション・トレース・ファイルを記録および保管できる。これらのファイルは、ライブ・ホスト接続の代わりに後から使用できる。記録されたトレース・ファイルを再生することにより、以下の操作を実行できる。ホスト端末機能を使用した、画面キャプチャー、画面イベント、および画面変換の作成。ホスト端末機能を使用した、マクロの作成およびテスト。Rational SDP ローカル・テスト環境を使用した、HATS アプリケーションのテスト。他のトレースおよびログと併せて行われる、ランタイム環境内で失敗したシナリオのトラブルシューティングの支援。

ホスト・シミュレーション・トレース (host simulation trace). ホスト・シミュレーション・トレース・ファイルは、ホスト画面とトランザクションを記録および保管する。これらのホスト画面とトランザクションは、ライブ・ホスト接続を使用する代わりに後から再生できる。トレース・ファイルは、ホスト端末機能を使用するかまたはランタイム環境内での実行中に記録できる。

ホスト端末 (host terminal). HATS Toolkit ツール。特定の HATS 接続と結合されたセッションであり、HATS 開発者は、画面の取り込み、画面カスタマイズの作成、およびマクロの記録に使用できる。

HTML. ハイパーテキスト・マークアップ言語 (Hypertext Markup Language)。

HTML ウィジェット (HTML widget). 「**ウィジェット (widget)**」を参照。

統合オブジェクト (Integration Object). 単一または一連のホスト画面との対話をカプセル化する Java Bean。統合オブジェクトはマクロで構成され、従来の (WSDL ベースの) Web サービス、RESTful Web サービス、または HATS EJB プロジェクトに組み込むことができる。統合オブジェクトは、リッチ・クライアント・プラットフォーム・アプリケーションでは使用できない。

相互運用性 (interoperability). コンピューターまたはプログラムが他のコンピューターまたはプログラムと連携する能力。

相互運用性ランタイム (interoperability runtime). バックエンド・ホストへの共通接続を管理するために、結合 HATS/WebFacing アプリケーションが使用する共通ランタイム。このランタイムは、WebFacing サーバーが戻すデータをアプリケーションの HATS 部分または WebFacing 部分のどちらで処理するかを決定する。

Java Platform, Enterprise Edition (Java EE). エンタープライズ・アプリケーションの開発およびデプロイのために Oracle によって定義された環境。 Java EE プラットフォームは、複数層の Web ベース・アプリケーションを開発するための機能を提供するサービス、アプリケーション・プログラミング・インターフェース (API)、およびプロトコルのセットで構成される。 (Oracle)

JavaServer Faces (JSF). Java で Web ベースのユーザー・インターフェースを作成するためのフレームワーク。Web 開発者は、ページ上に再利用可能な UI コンポーネントを配置し、コンポーネントをアプリケーション・データ・ソースに接続し、クライアント・イベントをサーバー・イベント・ハンドラーに接続することで、アプリケーションを構築できる。 (Oracle)

JavaServer Pages (JSP). サーバー・サイドのスクリプト記述テクノロジー。Web ページ (HTML ファイル) が提供されるときに、Java コードを動的にそのページに組み込んで実行し、クライアントに動的コンテンツを返すことができる。 (Oracle)

JavaServer Pages Standard Tag Library (JSTL). 共通構造化タスクのサポートを提供する標準タグ・ライブラリー。共通構造化タスクには、反復と条件、XML 文書の処理、国際化対応、構造化照会言語 (SQL) を使用したデータベース・アクセスなどがある。(Oracle)

JSF. 「**JavaServer Faces (JSF)**」を参照。

JSP. 「**JavaServer Pages (JSP)**」を参照。

JSR 168. Java ポートレット仕様は、ポータル環境で実行されるポートレットの集約、パーソナライゼーション、プレゼンテーション、およびセキュリティーの要件に対応する。Java ポートレット仕様のバージョン 1.0、Java Specification Request 168 (JSR 168) は、各種ベンダーが提供するポータル・サーバー間でのポートレットの互換性を有効にする標準を定義している。「**JSR 286**」を参照。

JSR 286. Java ポートレット仕様は、ポータル環境で実行されるポートレットの集約、パーソナライゼーション、プレゼンテーション、およびセキュリティーの要件に対応する。Java ポートレット仕様のバージョン 2.0、Java Specification Request 286 (JSR 286) は、ポートレット、リソース・サービス提供、およびその他の拡張機能間の調整など、バージョン 1.0 (JSR 168) の機能を拡張する標準を定義している。「**JSR 186**」を参照。

JSTL. 「**JavaServer Pages Standard Tag Library**」を参照。

キーボード・サポート (keyboard support). Web ブラウザーまたはリッチ・クライアント環境内でのアプリケーションの実行中にユーザーが物理キーボードを使用してホストと対話できるようにするために、開発者が利用する機能。開発者は、プロジェクトにホスト・キーパッドまたはアプリケーション・キーパッド (あるいはその両方) を含めるかどうかも決定する。キーパッドが含まれる場合、開発者は、クライアント・インターフェースに含めるキーおよびそれらのキーとキーパッドの表示方法も決定する。

キーパッド・サポート (keypad support). ユーザーがキーボード上の物理キーを押した場合と同様にホストと対話したり、アプリケーションに関連するタスク (印刷ジョブの表示や画面の最新表示など) を実行したりできるようにするために、開発者が利用する機能。「**アプリケーション・キーパッド (application keypad)**」および「**ホスト・キーパッド (host keypad)**」も参照。

リンク HATS/WebFacing プロジェクト (linked HATS/WebFacing project). 単一の HATS Web プロジェクトと単一の WebFacing プロジェクトをリンクさせて作成されたプロジェクト。HATS Web アプリケーションと WebFacing アプリケーションが相互運用されるエンタープライズ・アプリケーションを作成し、5250 バックエンド・ホストへの接続を共用することを目的とする。

Lotus Expeditor Client. Lotus Expeditor 製品のスタンドアロン・クライアント。ユーザーのマシンまたは開発マシンにインストールされる。

Lotus Notes クライアント (Lotus Notes Client). Lotus Notes 製品のスタンドアロン・クライアント。ユーザーのマシンまたは開発マシンにインストールされる。

マクロ (macro). マクロは、.hma ファイルに保管され、ホストとの対話を自動化する。これはコマンドをホストに送信し、入力フィールドにデータを入力し、ホストからデータを抽出し、ユーザーに代わって画面をナビゲートするために使用できる。

モデル 1 Web ページ (Model 1 Web pages). ユーザーに提示する情報、情報の表示方法を指定する書式設定タグ、およびページの表示順序を制御する論理を含む単一の JSP。(「**Struts Web ページ (Struts Web pages)**」と対比。)

ネットワーク・セキュリティー層 (network security layer). ユーザーを認証し、IBM Tivoli Access Manager などのネットワーク・リソースへのアクセスを許可するためのソフトウェア。

オペレーター情報域 (OIA) (Operator Information Area (OIA)). OIA は、ホスト・セッション画面下部の領域であり、セッション標識とメッセージが表示される。セッション標識は、ワークステーション、ホスト・システム、および接続に関する情報を示す。

パースペクティブ (perspective). Rational SDP ワークベンチで、そのワークベンチ内のリソースのさまざまな側面を表示するビューのグループ。 HATS パースペクティブはビューおよびエディターの集合であり、これにより開発者は HATS アプリケーションに属するリソースを作成、編集、表示、および実行できる。

プール (pooling). 「**接続プール (connection pool)**」を参照。

ポータル (portal). 統合 Web サイトのこと。リンク、コンテンツ、またはサービスなどの Web リソースのカスタマイズ・リストを動的に作成し、特定のユーザーのアクセス許可に基づいて、特定のユーザーが使用できる。

印刷サポート (print support). ホスト・セッションと関連付けるプリンター・セッションを指定し、ユーザーがホスト・アプリケーション印刷ジョブを表示、プリンターに送信、またはディスクに保管できるようにするために、開発者が利用する機能。印刷サポートは、デフォルト接続でのみ使用可能である。

プロファイル (Profile). リッチ・クライアント・プロジェクトの場合、「実行」と同じである。加えて最も多くの時間を必要とする操作を見つけ、繰り返されるアクションを識別し、冗長さを除去できる。この機能は、パフォーマンス分析に使用できる。そうすれば、アプリケーションをさらに詳しく理解できるようになる。

サーバーでプロファイル作成 (Profile on Server). Web プロジェクトの場合、「サーバーで実行」と同じである。加えて最も多くの時間を必要とする操作を見つけ、繰り返されるアクションを識別し、冗長さを除去できる。この機能をパフォーマンス分析に使用して、アプリケーションの理解を向上させるために役立てることができる。

プロジェクト (project). HATS リソースの集合 (成果物とも呼ばれる)。HATS Toolkit ウィザードを使用して作成され、HATS Toolkit エディターを使用してカスタマイズされる。これらのリソースは HATS アプリケーションとしてエクスポートされる。 HATS プロジェクトのタイプには、Web、ポートレット、EJB、リッチ・クライアントの各プロジェクトと、HATS Web (ポートレットおよび EJB を含む) アプリケーションの管理を目的とする HATS 管理コンソール・プロジェクトがあります。「**HATS プロジェクト (HATS project)**」または「**HATS EJB プロジェクト (HATS EJB project)**」を参照。

Rational Software Delivery Platform (Rational SDP). Eclipse オープン・ソース・プラットフォームをベースにした、e-business アプリケーションを開発するための一貫したツール・セットを提供する IBM ソフトウェア・プロダクト・ファミリー。

レンダリング・セット (rendering set). レンダリング・セットは、レンダリング項目の優先順位付きのリストを作成することにより構成される。各レンダリング項目は、1 つの特定の領域を定義する。その領域の内部で、指定されたホスト・コンポーネントが認識され、指定されたウィジェットを使用してレンダリングされる。

リソース (resource). HATS プロジェクトに組み込まれる複数のデータ構造のすべて。 HATS リソースには、テンプレート、画面イベント、変換、画面キャプチャー、接続、およびマクロが含まれる。他の Rational SDP プラグインでは、これらを「成果物」と呼ぶこともある。

RESTful Web サービス (RESTful Web service). 「**Web サービス、RESTful (Web service, RESTful)**」を参照。

リッチ・クライアント (rich client). クライアント環境内の Eclipse リッチ・クライアント・プラットフォーム上で実行するために設計されたプラグイン。展開先のプラットフォームのネイティブの外観および振る舞いを利用することにより、ユーザー・エクスペリエンスが向上するように設計されている。

実行 (Run). リッチ・クライアント・プロジェクトにおいて、Eclipse、Lotus Notes、または Lotus Expeditor Client インスタンスで HATS リッチ・クライアント・プロジェクトをテストできるようにする Rational SDP の一機能。このモードでは、ランタイム環境に展開されたランタイム設定 (runtime.properties ファイルにおいて定義) を変更およびテストできる。このモードでテストを行っている間にランタイム設定に対して行われた変更はすべて保持され、ランタイム環境に HATS アプリケーションを展開するときに有効になることに注意する必要がある。

| **サーバーで実行 (Run on Server).** Web プロジェクトにおいて、WebSphere Application Server で HATS Web プロジ
| ェクトおよびポートレット・プロジェクトを適宜テストできるようにする Rational SDP の一機能。このモードでは、
| ランタイム環境に展開されたランタイム設定 (runtime.properties ファイルにおいて定義) を変更およびテストできる。
| このモードでテストを行っている間にランタイム設定に対して行われた変更はすべて保持され、ランタイム環境に
| HATS アプリケーションを展開するときに有効になることに注意する必要がある。

ランタイム設定 (runtime settings). ランタイム環境に展開されたログ、トレース、および問題判別の設定 (runtime.properties ファイル内で定義)。

画面キャプチャー (screen capture). ホスト画面の XML 表現。.hsc ファイルに保管され、画面カスタマイズ、画面組み合わせ、変換、グローバル規則、またはマクロの作成またはカスタマイズに使用される。画面キャプチャーは、ホストに接続していない場合でも HATS プロジェクトの開発に使用できるため、有用である。 HATS 統合オブジェクトと Web サービス・サポートのコアであるマクロの作成にも有用である。

ビデオ端末 (VT) ホスト画面の画面キャプチャーは、Visual Macro Editor を使用して、およびプール構成時のチェックイン画面として、マクロを作成またはカスタマイズするために使用できる。これらの画面キャプチャーは、画面カスタマイズ、画面組み合わせ、変換、デフォルト・レンダリング、またはグローバル規則の作成には使用できない。

画面組み合わせ (screen combination). HATS 画面イベントのタイプの 1 つであり、連続する類似のホスト画面から出力データを収集し、それらを組み合わせ、単一の出力ページで表示するように設計されている。 .evnt ファイルに保管される画面組み合わせの定義内容には、組み合わされる開始および終了画面両方の一連の画面認識基準、画面間のナビゲート方法、各画面から収集されるデータの認識およびレンダリングに使用するコンポーネントとウィジェットが含まれる。

画面カスタマイズ (screen customization). ホスト画面の認識時に一連のアクションを実行するように設計された、画面イベントのタイプの 1 つ。 .evnt ファイルに保管される画面カスタマイズの定義内容には、ホスト画面を突き合わせるための一連の基準、およびホスト画面がそれらの基準と一致するときに実行されるアクションが含まれる。

画面イベント (screen event). 特定の画面認識基準と突き合わせることでホスト画面が認識されるときに起動される HATS イベント。画面カスタマイズと画面組み合わせの、2 つのタイプの画面イベントがある。

画面認識基準 (screen recognition criteria). HATS が 1 つ以上の画面と突き合わせるときに使用する一連の基準。ホストが画面を表示するときに、HATS は検索を実行し、現在のホスト画面が、プロジェクト内の画面イベントに対して定義された画面認識基準と一致するかを判別する。 HATS が一致するものを検出した場合、その画面イベントの既定のアクションが実行される。

画面認識基準はマクロの記録プロセスにも使用され、そのコンテキストでは「**記述子 (descriptors)**」と呼ばれることがある。

Secure Sockets Layer (SSL). 通信プライバシーを提供するセキュリティー・プロトコル。 SSL により、クライアント/サーバー・アプリケーションは、盗聴、改ざん、およびメッセージ偽造ができないように設計された方法で通信できる。 SSL は、Netscape Communications Corp. および RSA Data Security, Inc. により開発された。

ソース (source). HATS プロジェクトまたはそのリソースの 1 つを定義するマークアップ言語を含むファイル。さらに、各 HATS プロジェクトに含まれるフォルダーの名前。

SSL. 「**Secure Sockets Layer**」を参照。

標準ポートレット (standard portlets). Java Portlet Specifications JSR 168 または JSR 286 で定義されている標準ポートレット API に準拠するポートレット。「**JSR 168**」および「**JSR 286**」を参照。 .

Standard Widget Toolkit (SWT). Java 開発者向けの Eclipse ツールキット。基盤となるオペレーティング・システムのネイティブ・ウィジェットを使用する、移植可能な共通のユーザー・インターフェース API を定義する。

Struts Web ページ (Struts Web pages). Java Web アプリケーションを作成するための The Apache Software Foundation の Struts オープン・ソース・フレームワークを使用して構築された Web ページ。この Web ページ構築方

法では、クラス・ファイル (値を設定し、getter と setter を格納する)、入出力 JSP、および Web ダイアグラム (Web ページの流れと論理を表示する) を作成する。(「**モデル 1 Web ページ (Model 1 Web pages)**」と対比。)

SWT. 「**Standard Widget Toolkit**」を参照。

システム画面 (system screen). データ記述仕様 (DDS) ディスプレイ・ファイル・ソース・メンバーを使用できない IBM i 画面。システム画面は、WebFacing 化された IBM i 上のアプリケーションに固有の画面である。

テンプレート (template). .jsp ファイル (Web プロジェクトの場合) または .java ファイル (リッチ・クライアント・プロジェクトの場合) に保管されるテンプレート。アプリケーションの基本レイアウト、および色やフォントなどのスタイルを制御する。さらに、バナーやナビゲーション域などの、GUI に共通の領域の外観も定義する。

テキスト置換 (text replacement). HATS 画面変換時にホスト・システム上のテキストをイメージ、HTML コード、または他のテキストに変換するために使用される HATS 機能。

テーマ (theme). テーマでは、プロジェクトの共通の外観および振る舞いの特性一式がグループ化される。これらの属性は後から個別に変更できる。

転送 (transfer). アプリケーション EAR ファイルをサーバーにコピーすること。一般には FTP を使用する。

変換 (transformation). .jsp ファイル (Web プロジェクトの場合) または .java ファイル (リッチ・クライアント・プロジェクトの場合) に保管される変換。ホスト・コンポーネントを、GUI のウィジェットを使用して抽出および表示する方法を定義する。

変換接続 (transformation connection). 「**デフォルト接続 (default connection)**」を参照。

変換フラグメント (transformation fragment). 任意の変換において特定のパターンの出現個所をすべて置換するためのコンテンツを含む HATS リソース。

Unicode. 現代世界のいずれかの言語で書き記されたテキストの交換、処理、および表示をサポートする、汎用文字エンコード標準。多くの古典言語や、いくつかの言語の過去のテキストもサポートする。 Unicode 標準には、ISO 10646 で定義された 16 ビットの国際文字セットがある。

ユーザー (user). コンピューター・システムのサービスを使用する、人物、組織、プロセス、デバイス、プログラム、プロトコル、またはシステムのこと。

ユーザー・リスト (user list). HATS アプリケーションがホストまたはデータベースにアクセスするために使用できる、アカウント (ユーザー ID) に関する情報を記載したリスト。ユーザー・リストには、アカウントのユーザー ID、パスワード、および説明が記載されている。

UTF-8. Unicode 変換形式の 8 ビットのエンコード形式。既存の ASCII ベース・システムで使いやすくするために設計された。

Web アーカイブ (WAR) (Web archive (WAR)). Java EE 標準で定義された圧縮ファイル・フォーマット。Web アプリケーションのインストールおよび実行に必要なすべてのリソースを単一のファイルに保管する。

Web 高速ログオン (WEL) (Web Express Logon (WEL)). HATS の機能の 1 つであり、これによりユーザーは、ネットワーク・セキュリティー層で認証された一連の信任状を使用して複数のホストにログオンできる。 (「**ネットワーク・セキュリティー層 (network security layer)**」を参照。)

Web サービス (Web service). 標準ネットワーク・プロトコルを使用して、ネットワークを介して公開および呼び出すことができる、必要なものを完備した自己記述型のモジュラー・アプリケーション。

Web サービス、RESTful (Web service, RESTful). ステートレス・アーキテクチャーを使用し、関数呼び出しではなくリソースとして表示される Web サービス。Web サービス・リソースを識別するのに適切な書式の URI が使用され、アクティビティーを作成、取得、更新、および削除 (CRUD) するのに HTTP メソッド・プロトコルが使用され、メッセージ・フォーマットを定義するのに HTTP ヘッダー情報が使用される。

Web サービス、従来型、WSDL ベース (Web service, traditional, WSDL-based). 通常、データにタグを付けるのに XML が使用され、データの転送に SOAP が使用され、使用可能なサービスの記述に WSDL が使用され、どのサービスが使用可能かリストするのに UDDI が使用される Web サービス。

WebFacing フィーチャー (WebFacing feature). HATS Toolkit の IBM i 用 IBM WebFacing ツールのフィーチャー。 WebFacing フィーチャーを使用すると、IBM i データ記述仕様 (DDS) ディスプレイ・ファイル・ソース・メンバーを、既存の 5250 プログラム用 Web ベース・ユーザー・インターフェースに変換できる。

WebFacing アプリケーション (WebFaced application). HATS Toolkit の WebFacing フィーチャーによって生成される Web アプリケーション。

WebSphere. Web アプリケーションを実行するための e-business アプリケーションおよびミドルウェアを開発するためのツールを包含する IBM の商標名。 WebSphere Application Server の短縮名として使用される場合もある。これは、WebSphere プロダクト・ファミリーのランタイム部分を表す。

WebSphere Application Server. Web サーバー上で実行し、e-ビジネス・アプリケーションの展開、統合、実行、および管理に使用できる、Web アプリケーション・サーバー・ソフトウェア。 HATS アプリケーションは、エクスポートされてサーバーに転送されると、WebSphere Application Server のアプリケーションとして実行される。

WEL. 「**Web 高速ログオン (Web Express Logon (WEL))**」を参照。

ウィジェット (widget). 再使用可能なユーザー・インターフェース・コンポーネント。例えば、ボタン、スクロール・バー、制御域、またはテキスト編集域などで、キーボードまたはマウスから入力を受け取り、アプリケーションまたは他のウィジェットと通信できる。 HATS アプリケーションは、ホスト・コンポーネントをウィジェットに変換する。

ウィザード (wizard). アクティブな形式のヘルプであり、ユーザーに特定のタスクの各ステップを段階ごとに手引きする。

ワークベンチ (workbench). Rational SDP などの Eclipse ベースの製品におけるユーザー・インターフェースおよび統合開発環境 (IDE)。

XML. 「**Extensible Markup Language (XML)**」を参照。

Java に関する各種の定義は、Oracle からの許可を得て転載しました。

索引

日本語, 数字, 英字, 特殊文字の順に配列されています。なお, 濁音と半濁音は清音と同等に扱われています。

［ア行］

アイコン 384
アクション
  アプリケーション・イベント 3
  一時停止 194
  画面イベント 3
  キーを送信 193
  グローバル変数を除去 186
  グローバル変数を設定 186
  グローバル変数を送信 187
  グローバル変数を抽出 183
  切断 194
  説明 3
  データを挿入 185
  ビジネス・ロジックを実行 183
  変換を適用 183
  マクロを実行 192
  マクロ・トランザクションを実行 193
  URL に転送 190
  URL または SWT コンポジットを表示 189
  URL を表示 189
「アクション」タブ
  画面イベント 181
  画面カスタマイズ 181
  画面組み合わせ 181
アクセント付き文字
  コード・ページ 937 509
圧縮
  HTTP 142
アプリケーション
  イベント 125
  開発手順 11
  キーパッド 379
  Web
    編成 35
「アプリケーション」ビュー 94
  ポップアップ・メニュー 95
アプリケーション処理
  HATS 9
アプリケーション・イベント 125
  アクション 3
  説明 2
アプリケーション・キーパッド 379
アプリケーション・キーパッド *(続き)*
  設定 121
  挿入 219
    カスタム 219
    個別キー 220
    デフォルト 219
アプリケーション・キーパッドを挿入 219
アプレット
  自動切断および最新表示 136
アラビア語コード・ページ
  双方向 534
一時画面
  マクロ処理 396
イベント
  アプリケーション 125
  エラー 452
  画面 172
  警告 452
  説明 2
  ロギング 452
「イベント」タブ
  プロジェクトの変更 124
「印刷」タブ
  接続 162
印刷サポート
  「印刷」タブ 162
  印刷画面 97
  概要 416
  構成 417
  使用可能化 416
  説明 8
  定義 417
  ユーザー 421
  リッチ・クライアント
    プリファレンス 99
  3270 接続の場合 417
  5250 接続の場合 420
  Host On-Demand PDT コンパイラー 419
インストール
  HATS Web アプリケーション 40
  HATS リッチ・クライアント・アプリケーション 86
    Eclipse RCP 86
    Lotus Expeditor Client 91
    Lotus Notes 87
インポート
  リッチ・クライアント・プロジェクト 73
  Web プロジェクト 38
ウィザード
  アプリケーション・キーパッドを挿入 219
  オペレーター情報域を挿入 218
  格納済み画面を挿入 220
  グローバル変数を挿入 218
  すべてのフィールドに抽出アクションを追加 393
  すべてのフィールドにプロンプト・アクションを追加 392
  すべてのホスト・コンポーネントを挿入 220
  タブ付きフォルダーを挿入 216
  抽出を追加 389
  デフォルト・レンダリングを挿入 215
  デフォルト・レンダリングを編集 215
  プロンプトの追加 387
  ホスト・キーパッドを挿入 218
  ホスト・コンポーネントを挿入 213
  ホスト・コンポーネントを編集 215
  マクロの記録 385
  マクロ・キーを挿入 217
  ループの記録 392
ウィジェット
  概要 4
  カレンダー 262
  グラフ (水平棒、折れ線、垂直棒) 287
  コンボ 271
  サブファイル (チェック・ボックス) 314
  サブファイル (ドロップダウン) 321
  サブファイル (ポップアップ) 329
  スクロール・バー (ENPTUI) 313
  設定 121, 258
  ダイアログ 274
  チェック・ボックス 269
  ツールバー 347
  テーブル 337
  テキスト入力 343
  ドロップダウン (選択) 279
  ドロップダウン (データ入力) 276
  フィールド 281
    マイグレーション 21
  ボタン 258
    マイグレーション 20
  ボタン・テーブル 260
  ポップアップ 300
  ラジオ・ボタン (項目選択) 308
  ラジオ・ボタン (選択) 310
  ラジオ・ボタン (データ入力) 305

© Copyright IBM Corp. 2003, 2015

ウィジェット *(続き)*
ラベル 292
リスト 297
リンク 293
リンク (項目選択) 295
Dojo 351
拡張グリッド 357
検証テキスト・ボックス 362
コンボ・ボックス 352
テキスト・ボックス 361
日付テキスト・ボックス 354
フィルター操作選択 358
ウィジェットの位置合わせ
双方向 522
ウィジェットの方向
双方向 521
上書きモード 145
エクスポート
ポートレット 466
リッチ・クライアント・プロジェクト 73
HATS ランタイム・フィーチャー 83
HATS リッチ・クライアント・プロジェクト 81
Web プロジェクト 37
Java EE 39
エディター
接続 158
マクロ 382
エラー・イベント 127, 452
エンコード
設定 509
UTF-8 509
エンコードの設定 509
オペレーター情報域 122
設定 122
挿入 218
オペレーター情報域を挿入 218
オンライン・トランザクション処理 (OLTP) 473

[カ行]

カーソル位置 144
開始イベント 125
「開始画面」タブ
画面イベント 175
画面カスタマイズ 175
画面組み合わせ 175
開始状態ラベル 406
開発手順
アプリケーション 11
「概要」タブ
画面イベント 175
画面カスタマイズ 175
画面組み合わせ 175
「概要」タブ *(続き)*
接続 158
プロジェクトのテスト 108
プロジェクトの変更 107
プロジェクト・テーマ 107
マクロ・エディター 382
拡張
エディター 394
「拡張」タブ
接続 159
拡張エディター 394
拡張グリッド Dojo ウィジェット 357
拡張レンダリング 114
格納済み画面
挿入 220
格納済み画面を挿入 220
カスタム・テーブルのサポート 514
画面
結合 414
複数、入力 416
複数アプリケーションの出力 415
不連続出力データ 415
連続出力データ 415
使用禁止 198
ロック 198
画面イベント
アクション 3
「アクション」タブ 181
一時停止 194
色基準 178
カーソル位置基準 176
「開始画面」タブ 175
「概要」タブ 175
カスタム基準 179
キーを送信 193
「グローバル規則」タブ 194
グローバル変数基準 177
グローバル変数を除去 186
グローバル変数を設定 186
グローバル変数を送信 187
グローバル変数を抽出 183
作業 172
「終了画面」タブ 180
条件基準 179
切断 194
説明 3
「ソース」タブ 197
「次の画面」タブ 197
データを挿入 185
「テキスト置換」タブ 195
テキスト・ストリング位置基準 177
「ナビゲーション」タブ 180
ビジネス・ロジックを実行 183
フィールド基準 175
「変換を適用」アクション 183
編集 175
画面イベント *(続き)*
ポートレットのプレプレゼンテーション 194
マクロを実行 192
マクロ・トランザクションを実行 193
優先順位 124
領域と値の比較の基準 178
領域を領域と比較する基準 178
「レンダリング」タブ 179
URL に転送 190
URL または SWT コンポジットを表示 189
URL を表示 189
画面カスタマイズ
「アクション」タブ 181
一時停止 194
色基準 178
ウィザード 173
カーソル位置基準 176
「開始画面」タブ 175
「概要」タブ 175
カスタム基準 179
キーを送信 193
「グローバル規則」タブ 194
グローバル変数基準 177
グローバル変数を除去 186
グローバル変数を設定 186
グローバル変数を送信 187
グローバル変数を抽出 183
作業 172
条件基準 179
切断 194
説明 3
「ソース」タブ 197
「次の画面」タブ 197
データを挿入 185
「テキスト置換」タブ 195
テキスト・ストリング位置基準 177
ビジネス・ロジックを実行 183
フィールド基準 175
「変換を適用」アクション 183
編集 175
ポートレットのプレプレゼンテーション 194
マクロを実行 192
マクロ・トランザクションを実行 193
優先順位 124
領域と値の比較の基準 178
領域を領域と比較する基準 178
URL に転送 190
URL または SWT コンポジットを表示 189
URL を表示 189
画面キャプチャー
説明 7
双方向 520

画面組み合わせ 414
「アクション」タブ 181
一時停止 194
色基準 178
ウィザード 174
カーソル位置基準 176
「開始画面」タブ 175
「概要」タブ 175
カスタム基準 179
キーを送信 193
「グローバル規則」タブ 194
グローバル変数基準 177
グローバル変数を除去 186
グローバル変数を設定 186
グローバル変数を送信 187
グローバル変数を抽出 183
作業 172
「終了画面」タブ 180
条件基準 179
切断 194
説明 3
「ソース」タブ 197
「次の画面」タブ 197
データを挿入 185
「テキスト置換」タブ 195
テキスト・ストリング位置基準 177
「ナビゲーション」タブ 180
ビジネス・ロジックを実行 183
フィールド基準 175
「変換を適用」アクション 183
編集 175
ポートレットのプレプレゼンテーション 194
マクロを実行 192
マクロ・トランザクションを実行 193
優先順位 124
領域と値の比較の基準 178
領域を領域と比較する基準 178
「レンダリング」タブ 179
URL に転送 190
URL または SWT コンポジットを表示 189
URL を表示 189
画面スキップ・マクロ
説明 381
画面設定
リファレンス 567
画面認識基準
色 178
オプション 176
カーソル位置 176
カスタム 179
グローバル変数 177
条件 179
説明 3
追加の基準 177
画面認識基準 *(続き)*
テキスト・ストリング位置 177
反転一致 177
非オプション 176
フィールド 175
領域と値の比較 178
領域と領域の比較 178
画面の向き
双方向 525
カレンダー・ウィジェット 262
管理
接続 157
ポートレット 467
管理コンソール
機能 441
役割 445
キーパッド
アプリケーション 121, 379
外観 423
変更 423
ホスト 122, 221
Cascading Style Sheet 423
キーパッドの非表示 393
キーパッドの表示 393
キーパッド・サポート
説明 9
キーボード
再マップ 514
キーボードの再マップ 428
リッチ・クライアント 434
Web 431
キーボード・サポート 129
概要 423
キー・コード値 430
キー・マッピング 425
再マップ 428
リッチ・クライアント 434
Web 431
使用可能化 423
設定 129
説明 9
定義 423
ニーモニック・キーワード 428
マッピング 425
ユーザー 424
5250 フィールド・キー・サポート 426
キーボード・マッピング
マイグレーション 23
キー・コード値 430
「基本」タブ
接続 158
クライアント設定 142
クライアント・ロケール 130
設定 130
クラスター環境
HATS 構成 43
クラス・ローダー・ポリシー
構成 42
グラフ (水平棒、折れ線、垂直棒) ウィジェット 287
グローバル規則
説明 5
双方向 524
プロジェクト設定 116
マイグレーション 21
「グローバル規則」タブ
画面イベント 194
画面カスタマイズ 194
画面組み合わせ 194
グローバル変数
共用 399
使用 397
除去 186
設定 186
説明 6
送信 187
挿入 218
双方向 523
取り出し中 183
名前変更 401
マイグレーション 21
ローカル 399
グローバル変数のオーバーライド 141
設定 141
ポートレット 460
リッチ・クライアント 96
グローバル変数を挿入 218
結合
画面 414
複数、入力 416
複数アプリケーションの出力 415
不連続出力データ 415
連続出力データ 415
言語、選択 447
言語コード 506
言語サポート 505
エンコードの設定 509
キーボードおよびディスプレイの再マップ 514
言語コード 506
コード・ページ 506
コード・ページ 1388 510
コード・ページ 1390 および 1399 510
コード・ページ 937 509
GB18030 510
JIS2004 510, 511
Unicode サポート 509
検証テキスト・ボックス Dojo ウィジェット 362

コード・ページ 506
コード・ページ 1388
GB18030 510
コード・ページ 1390 および 1399
ホスト・コード・マッピング 510
コード・ページ 937
アクセント付き文字 509
コード・ページのサポート
双方向 515
更新サイト
リッチ・クライアント 84
構成
印刷サポート 417
項目選択コンポーネント 238
コピー
プロジェクト・リソース 72
コマンド行コンポーネント 223
コンテンツ・アシスタンス
マクロ 147, 395
コンパクト・レンダリング 113
コンポーネント
項目の選択 238
コマンド行 223
サブファイル 244
設定 120, 223
選択リスト 242
ダイアログ 224
テーブル 252
テーブル (可視) 255
テーブル (フィールド) 254
テキスト 257
入力フィールド 234
ヒント付き入力フィールド 236
ファンクション・キー 231
フィールド 230
ライト・ペン (アテンション) 240
ライト・ペン (選択) 241
ENPTUI 228
HTML DDS キーワード 233
URL 257
コンポーネントおよびウィジェット 221
設定 120, 221
説明 221
マッピング 364
コンボ・ウィジェット 271
コンボ・ボックス Dojo ウィジェット 352

[サ行]

サーバー管理
使用 448
サーバーで実行
説明 7
サーバーでデバッグ
説明 7
サーバーでプロファイル作成
説明 8
サーバーの考慮事項
ディスク・スペース 35
デプロイメント 36
トレース 36
ライセンス追跡 36
ロギング 36
サーバー・モジュールの可視性
構成 42
再マップ
キーボードおよびディスプレイの文字 514
作成
接続 157
サブファイル (チェック・ボックス) ウィジェット 314
サブファイル (ドロップダウン) ウィジェット 321
サブファイル (ポップアップ) ウィジェット 329
サブファイル・コンポーネント 244
サポート
Unicode 509
実行
説明 8
自動拡張 143
マイグレーション 19
自動最新表示 134
自動切断 134
自動切断および最新表示
クライアント・プル (AJAX) 134
サーバー・プッシュ (アプレット) 136
Web アプリケーション
設定 134
「終了画面」タブ
画面イベント 180
画面組み合わせ 180
上級者向けトピック
端末の表示 454
使用禁止
画面 198
使用禁止画面 198
状態ラベル、開始 406
状態ラベル、停止 406
使用中のページ 146
使用中のページを有効にする 146
信任状マッパー
選択 487
スクロール・バー (ENPTUI) ウィジェット 313
スタイル・シート
使用 374
テンプレート 375
スタイル・シート、カスケード 423
スプレッドシート
スプレッドシート・ファイルの提供 319, 326, 334, 342
すべてのフィールドに抽出アクションを追加 393
すべてのフィールドにプロンプト・アクションを追加 392
すべてのホスト・コンポーネント
挿入 220
すべてのホスト・コンポーネントを挿入 220
正規表現
「テキスト置換」タブ 195
セキュリティー
Java 2 セキュリティー 497
Kerberos 496
Kerberos を使用可能にする 168
SSH 480
SSL 477
SSL マイグレーション 25
SSL を使用可能にする 166
Web 高速ログオンを使用可能にする 168
WEL 481
WEL マイグレーション 25
「セキュリティー」タブ
接続 166
接続
「印刷」タブ 162
エディター 158
「概要」タブ 158
「拡張」タブ 159
管理 157, 449
「基本」タブ 158
作成 157
「セキュリティー」タブ 166
「ソース」タブ 172
「プール」タブ 168
プロジェクトの変更 109
「マクロ」タブ 169
「ユーザー・リスト」タブ 170
接続イベント 126
接続エディター 158
接続設定
screenRecoUseOIASnapshot 396
接続パラメーターのオーバーライド 130
設定 130
ポートレット 460
リッチ・クライアント 95
接続プール
HATS 管理コンソール 449
切断イベント 128
設定
キーボード・サポート 129
クライアント・ロケール 130

設定 *(続き)*
グローバル変数のオーバーライド 141
接続パラメーターのオーバーライド 130
disconnectOnClose 133
keepOutputTogether 232
先行入力サポート 147
先行入力サポートを使用可能にする 147
選択リスト・コンポーネント 242
「ソース」タブ
画面イベント 197
画面カスタマイズ 197
画面組み合わせ 197
接続 172
テンプレート 376
プロジェクトの変更 149
変換 209
マクロ・エディター 394
挿入、HATS アプリケーションへの転送の 414
双方向
アラビア語コード・ページの機能 534
ウィジェットの位置合わせ 522
ウィジェットの方向 521
画面キャプチャー 520
画面の向き 525
グローバル規則 524
グローバル変数 523
ソフトウェア環境 518
追加、HATS ファイルへの 538
テキスト置換 523
統合オブジェクト 529
ホスト端末の操作 518
ホスト・コンポーネント 520
右から左の印刷 528
リッチ・クライアント・アプリケーションの位置合わせ 518
IBM i 5250 Unicode サポート 534
VT 表示オプション 525, 531
Web サービス 517, 530
双方向のサポート
概要 515
属性
enableScrRev
session タグ 538
「その他」タブ
プロジェクトの変更 129
ソフトウェア環境
双方向 518

[タ行]

ダイアログ・ウィジェット 274
ダイアログ・コンポーネント 224
代替レンダリング 114
対話式 CPW 473
タブ順序 143
タブ付きフォルダー
挿入 216
「タブ付きフォルダーを挿入」ウィザード 216
端末
表示 7
ホスト 7
端末の表示
説明 7
テストとデバッグ 454
HATS 管理コンソール 453
チーム共用
プロジェクト 35, 71
チェーニング
使用するアプリケーションのデバッグ 408
統合オブジェクト
概要 405
どのようなときに使用するかの判断 405
チェック・ボックス・ウィジェット 269
抽出アクションを追加 389
抽出マクロ
説明 381
ツールバー
設定 121
ツールバー・ウィジェット 347
追加、HATS ファイルへの
双方向 538
追加のキーパッド・ボタンの追加
変換 436
「次の画面」タブ
画面イベント 197
画面カスタマイズ 197
画面組み合わせ 197
データ
挿入 185
テーブル (可視) コンポーネント 255
テーブル (フィールド) コンポーネント 254
テーブル・ウィジェット 337
テーブル・コンポーネント 252
テーマ
プロジェクト 107
定義
印刷サポート 417
キーボード・サポート 423
停止イベント 128
停止状態ラベル 406
ディスプレイ
再マップ 514
テキスト
選択 145
テキスト OIA の表示 393
テキスト置換
説明 5
双方向 523
プロジェクト設定 120
「テキスト置換」タブ
画面イベント 195
画面カスタマイズ 195
画面組み合わせ 195
正規表現 195
テキスト入力ウィジェット 343
テキスト・コンポーネント 257
テキスト・ボックス Dojo ウィジェット 361
「デザイン」タブ
テンプレート 373
変換 206
テスト
サーバーで実行 7
サーバーでデバッグ 7
サーバーでプロファイル作成 8
実行 8
デバッグ 8
プロジェクト 108
プロファイル 8
テスト・モード
マイグレーション 25
デバッグ
説明 8
デフォルト・レンダリング
説明 5
挿入 215
プロジェクト設定 111
編集 215
「デフォルト・レンダリングを挿入」ウィザード 215
「デフォルト・レンダリングを編集」ウィザード 215
デプロイ
HATS リッチ・クライアント・アプリケーション 80
展開
HATS Web アプリケーション 38
転送、HATS アプリケーションへの
挿入 414
テンプレート
ウィザード 370
概要 5
作成ウィザード 370
使用 367
スタイル・シート 375
説明 5
「ソース」タブ 376
「デザイン」タブ 373
「プレビュー」タブ 376
分割タブ 376
編集 372

テンプレート *(続き)*
リッチ・クライアント 377
Web 372
例 368
「テンプレート」タブ
プロジェクトの変更 110
統合オブジェクト
作成 404
作成、モデル 1 Web ページの 409
作成、JSF Web ページの 411
作成、Struts Web ページの 410
出力の挿入 413
使用 402
説明 6
双方向 529
チェーニング 405, 406
入力の挿入 413
Web ページの構築 409
トラブルシューティング
リッチ・クライアント
プリファレンス 99
リッチ・クライアント・アプリケーション 93
Web アプリケーション 450
トレース
オプションの設定 452
クラスター環境 43
サーバーの考慮事項 36
トレース設定
オプション 561
ホスト・シミュレーション 565
リッチ・クライアント・アプリケーション 93
Host On-Demand 563
runtime.properties ファイル 557
Web アプリケーション 450
トレース・ファイル 560
ドロップダウン (選択) ウィジェット 279
ドロップダウン (データ入力) ウィジェット 276

[ナ行]

「ナビゲーション」タブ
画面イベント 180
画面組み合わせ 180
ニーモニック
キーワード 428
ニーモニック・キーワード 428
入力フィールド・コンポーネント 234
ネットワーク・セキュリティー・プラグイン 485

[ハ行]

バージョン
インディケーター 13
パターン
レンダリング 203
バックアップ
Web プロジェクト 37
バックアップ・ファイル
マイグレーション 19
パフォーマンス
HTTP 圧縮 142
反転一致基準 177
ビジネス・ロジック
実行 183
説明 6
ビジュアル・マクロ・エディター 381
日付テキスト・ボックス Dojo ウィジェット 354
非同期更新
アプレット 136
リッチ・クライアント
設定 134
ヒント 396
マクロ 396
ヒント付き入力フィールド・コンポーネント 236
「プール」タブ
接続 168
プール定義 450
ファイル
トレース 560
.ear 35
.war 35
ファンクション・キー・コンポーネント 231
フィールド・ウィジェット 281
マイグレーション 21
フィールド・キー・サポート
5250 426
フィールド・コンポーネント 230
不一致画面イベント 127
フィルター操作選択 Dojo ウィジェット 358
フォーカスされたテキストをすべて選択する 145
DBCS 546
複数の画面
結合 414
ブランク画面イベント 126
プリファレンス
リッチ・クライアント 99
印刷 99
トラブルシューティング 99
HATS Toolkit 149
Rational SDP 155
「プレビュー」タブ
テンプレート 376
変換 209
プロキシー・サーバー
カレンダー・ウィジェット 262
グラフ・ウィジェット 288
テキスト置換 197
ポップアップ・ウィジェット 300
HATS 構成 44
プロジェクト
設定 107
説明 2
チーム共用 35, 71
変更 107
リソースのコピー 72
リッチ・クライアント
移動 72
インポート 73
エクスポート 73
Web
移動 36
インポート 38
エクスポート 37
作成 33
バックアップ 37
.ear ファイル 36
プロジェクト設定
アプリケーション・キーパッド 121
ウィジェット 121
オペレーター情報域 122
グローバル規則 116
コンポーネント 120
コンポーネントおよびウィジェット 120
ツールバー 121
テキスト置換 120
デフォルト・レンダリング 111
ホスト・キーパッド 122
プロジェクトの変更 107
「イベント」タブ 124
「概要」タブ 107
接続 109
「ソース」タブ 149
「その他」タブ 129
「テンプレート」タブ 110
「レンダリング」タブ 110
プロジェクト・テーマ 107
プロファイル
説明 8
「プロンプトおよび抽出」タブ
マクロ・エディター 394
プロンプト・アクションを追加 387
プロンプト・マクロ
説明 381
分割タブ
テンプレート 376

分割タブ *(続き)*
変換 209
変換
ウィザード 213
作業 201
作成ウィザード 203
説明 4
「ソース」タブ 209
追加のキーパッド・ボタンの追加 436
適用 183
「デザイン」タブ 206
パターン 203
プレビュー 220
「プレビュー」タブ 209
分割タブ 209
編集 205
リッチ・クライアント 209
Web 205
ホスト・キーパッド 221
「変換」ビュー 97
ポップアップ・メニュー 98
編集
画面イベント 175
画面カスタマイズ 175
画面組み合わせ 175
変数
グローバル 6
ポートレット
エクスポート 466
管理 467
考慮事項 468
作業 460
作成 457
新規作成 458
制限事項 468
通信 461
テスト 466
パラメーターのオーバーライド 460
標準
WEL 470
プレプレゼンテーション 194
ブロック区切り文字を追加 194
連携 461
HATS 457
HATS プロジェクトからの生成 458
Web Services for Remote Portlets 465
WSRP 465
ポートレットの設定 148
ホスト
キーパッド 221
ホストおよびアプリケーション・サーバーの選択 448
ホスト画面のプレビュー 393
ホスト端末
説明 7
双方向 518
ホスト端末 *(続き)*
マクロの処理 383
ホストの印刷サポート
参照： 印刷サポート
ホスト・キーパッド
設定 122
挿入 218
カスタム 219
個別キー 219
デフォルト 218
ホスト・キーパッドを挿入 218
ホスト・キーボード・サポート
参照： キーボード・サポート
ホスト・コンポーネント
概要 4
設定 223
挿入 213
双方向 520
編集 215
「ホスト・コンポーネントを挿入」ウィザード 213
「ホスト・コンポーネントを編集」ウィザード 215
ホスト・シミュレーション 436
ウィザード 437
エディター 438
実行時の記録 439
トレース設定 565
リッチ・クライアント・ランタイム・トレースを使用可能にする 101
Web アプリケーション・ランタイム・トレースを使用可能にする 454
ボタン・ウィジェット 258
マイグレーション 20
ボタン・テーブル・ウィジェット 260
ポップアップ・ウィジェット 300
ポップアップ・メニュー
「アプリケーション」ビュー 95
「変換」ビュー 98

[マ行]

マイグレーション
キーボード・マッピング 23
グローバル規則 21
グローバル変数 21
自動拡張 19
テスト・モード 25
特別な考慮事項 19
バックアップ・ファイル 19
フィールド・ウィジェット 21
プロジェクトのインポート 17
変換 19
ボタン・ウィジェット 20
ライセンス設定 24
ランタイムの使用可能化 24
マイグレーション *(続き)*
リッチ・クライアント 23
ワークスペース・プロジェクト 18
EJB Access Bean 20
HATS 15
Host Publisher 26
HTTP 圧縮 22
IBM License Metric Tool 22
IBM Tivoli License Compliance Manager 22
ITLM 22
J2EE 30
SSL 25
TLCM 22
URL ホスト・コンポーネント 25
WEL 25
マクロ
一時画面の処理 396
インポート 395
エクスポート 396
エディター 382
「概要」タブ 382
「ソース」タブ 394
「プロンプトおよび抽出」タブ 394
エラー 395
拡張エディター 394
画面スキップ 381
キーパッドの非表示 393
キーパッドの表示 393
記録 385
組み込み 380
コンテンツ・アシスタンス 147, 395
実行 192
すべてのフィールドに抽出アクションを追加 393
すべてのフィールドにプロンプト・アクションを追加 392
説明 6
抽出 381, 389
テキスト OIA の表示 393
ビジュアル・エディター 381
ヒント 396
プロンプト 381, 387
無限ループの回避 396
ループの記録 392
「マクロ」タブ
接続 169
マクロのインポート 395
マクロのエクスポート 396
「マクロの記録」ウィザード 385
マクロ・キー
挿入 217
マクロ・キーを挿入 217

マッピング
コンポーネントおよびウィジェット 364
右から左の印刷
双方向 528
未変更フィールド
送信の抑制 143
無限ループ
回避 396
モデル 1 Web ページ
統合オブジェクト 409
モバイル装置
カーソル位置 285, 302, 319, 326, 334, 340, 345
カーソル位置決め 59
考慮事項 54
Android 63
iPad デバイス 62
コンパクト・レンダリング 113
サブファイル列およびテーブル列の配置 55, 316, 323, 331, 339
制限事項 54
フリー・レイアウト・テーブルのプリファレンス 154
DBCS 549
HATS アプリケーションの開発 53
問題判別 14
リッチ・クライアント・アプリケーション 93
Web アプリケーション 450

[ヤ行]

役割
HATS 管理コンソール 445
矢印キー・ナビゲーション 146
矢印キー・ナビゲーションを使用可能にする 146
ユーザー
印刷サポート 421
キーボード・サポート 424
ユーザー定義文字 550
ユーザー・リスト
管理 450
「ユーザー・リスト」タブ
接続 170
優先順位
画面イベント 124
画面カスタマイズ 124
画面組み合わせ 124

[ラ行]

ライセンス
管理 449
ライセンス *(続き)*
licenseFile タグ (runtime.properties) 558
licenseHardLimit タグ (runtime.properties) 558
licenseTracking タグ (runtime.properties) 558
numLicenses タグ (runtime.properties) 557
ライセンス管理
HATS 管理コンソール 449
ライセンス設定
パネル 449
マイグレーション 24
ランタイムの使用可能化 39, 80
リッチ・クライアント 81
リッチ・クライアント・アプリケーション 93
runtime.properties ファイル 557
ライセンス追跡
クラスター環境 43
サーバーの考慮事項 36
ライト・ペン (アテンション) コンポーネント 240
ライト・ペン (選択) コンポーネント 241
ラジオ・ボタン (項目選択) ウィジェット 308
ラジオ・ボタン (選択) ウィジェット 310
ラジオ・ボタン (データ入力) ウィジェット 305
ラベル
開始 406
停止 406
ラベル・ウィジェット 292
ランタイムの使用可能化
マイグレーション 24
ライセンス設定 39, 80
リスト・ウィジェット 297
リッチ・クライアント 63
「アプリケーション」ビュー 94
ポップアップ・メニュー 95
アプリケーション・フィーチャーのエクスポート 81
管理 93
グローバル変数のオーバーライド 96
更新サイト 84
接続パラメーターのオーバーライド 95
ツールバー設定 121
トラブルシューティング 93
トレース 93
バージョン番号の更新 92
非同期更新
設定 134
リッチ・クライアント *(続き)*
プリファレンス 99
印刷 99
トラブルシューティング 99
プロジェクト 70
「変換」ビュー 97
ツールバー 97
テンプレートおよび変換領域 98
ビューのクローズ 98
ホスト・キーパッド 98
ポップアップ・メニュー 98
ランタイム状況表示パネル 98
OIA 状況域 98
マイグレーション 23
ライセンス設定 81, 93
ロギング 93
ワークステーション ID のプロンプト表示 99
HATS RCP ランタイム拡張プロジェクト 69
HATS アプリケーションのインストール 86
Eclipse RCP 86
Lotus Expeditor Client 91
Lotus Notes 87
HATS アプリケーションの開発 66
HATS アプリケーションのテスト 74
HATS アプリケーションのデプロイ 80
HATS の考慮事項 103
HATS の制限 103
HATS ランタイム・フィーチャーのエクスポート 83
LU 名のプロンプト表示 99
リッチ・クライアント・アプリケーションの位置合わせ
双方向 518
リッチ・クライアント・プラットフォーム
HATS 63
リモート・ポートレットの Web サービス 465
リンク (項目選択) ウィジェット 295
リンク・ウィジェット 293
ループ・アクションの記録 392
連携ポートレット 461
レンダリング
拡張 114
コンパクト 113
代替 114
パターン 203
HTML テーブル 115
「レンダリング」タブ
画面イベント 179
画面組み合わせ 179
プロジェクトの変更 110
レンダリング・セット 111

ロギング
クラスター環境 43
サーバーの考慮事項 36
ログオン・マクロの作成
WEL 484
ログ設定
リッチ・クライアント・アプリケーション 93
runtime.properties ファイル 557
Web アプリケーション 450
ログ・ファイル
オプションの制御 452
表示 452
ロジック
ビジネス 6
ロック
画面 198

[ワ行]

ワークステーション ID
設定 160
ワークステーション ID のプロンプト表示
リッチ・クライアント 99

[数字]

2 バイト文字セット 541
上書きモード 546
画面認識 547
キーボードおよびディスプレイの文字再マップ 514, 550
コピー/貼り付け操作の防止 542
サブファイル・コンポーネント 547
自動拡張 546
シフトアウト/シフトインに関する考慮事項 544
選択リスト・コンポーネント 547
データ・タイプの検査 542
テーブル・コンポーネント 547
統合オブジェクトの作成 549
フィールド長の検査 542
フィールドのサポート 541
フォーカスされたテキストをすべて選択する 546
プロジェクト設定エディター 544
プロジェクト・テーマの設定 544
プロジェクト・レベルのレンダリング 544
マクロのエクスポート 549
マクロの記録 549
モバイル装置アプリケーション 549
ユーザー定義文字 550
AutoIME 切り替え 543
2 バイト文字セット *(続き)*
DBCS での最大長の除去 544, 545
Input Method Editor (IME) 543
SBCS での最大長の除去 548
UDC 550
5250 Unicode サポート 509
5250 フィールド・キー・サポート 426
5250W
HATS 接続 158, 473

A

adminPortNum (runtime.properties) 557
AdvancedIOErrorPage
BasicIOErrorPage.jsp 412
Android デバイス
考慮事項 63
Apache Geronimo
HATS アプリケーションの開発 45

B

BasicIOErrorPage.jsp
AdvancedIOErrorPage 412
BasicIOErrorPage.jsp および AdvancedIOErrorPage.jsp 412
Bluemix
HATS アプリケーションの開発 49
BMS マップ・セット
インポート 199

C

Cascading Style Sheet 423
connection
説明 5
CPW 473
Credential Mapper プラグイン 485

D

DBCS 541
上書きモード 546
画面認識 547
キーボードおよびディスプレイの文字再マップ 514, 550
コピー/貼り付け操作の防止 542
サブファイル・コンポーネント 547
自動拡張 546
シフトアウト/シフトインに関する考慮事項 544
選択リスト・コンポーネント 547
データ・タイプの検査 542
テーブル・コンポーネント 547
統合オブジェクトの作成 549
DBCS *(続き)*
フィールド長の検査 542
フィールドのサポート 541
フォーカスされたテキストをすべて選択する 546
プロジェクト設定エディター 544
プロジェクト・テーマの設定 544
プロジェクト・レベルのレンダリング 544
マクロのエクスポート 549
マクロの記録 549
モバイル装置アプリケーション 549
ユーザー定義文字 550
AutoIME 切り替え 543
DBCS 最大長の除去 545
DBCS での最大長の除去 544
Input Method Editor (IME) 543
SBCS 最大長の除去 548
UDC 550
disconnectOnClose
パラメーター 133
Dojo
ウィジェット
拡張グリッド 357
検証テキスト・ボックス 362
コンボ・ボックス 352
設定 351
テキスト・ボックス 361
日付テキスト・ボックス 354
フィルター操作選択 358
Dojo ウィジェット 409, 413

E

EAR
ファイル 35
EJB Access Bean
マイグレーション 20
EJB プロジェクト
SSL セキュリティー 166
enableScrRev 属性
session タグ 538
ENPTUI コンポーネント 228

G

GB18030
考慮事項 510
Geronimo
HATS アプリケーションの開発 45

H

HATS
アプリケーション開発 11

HATS *(続き)*
アプリケーション処理 9
ウィジェット 258
概要 i
グローバル変数 397
処理順序 9
プロジェクト
変更 107
ホスト・コンポーネント 223
macros
組み込み 380
WebFacing 473
WebFacing 相互運用性 474
HATS 管理コンソール
アクセス 447
機能 441
接続の管理 449
接続プール 449
端末の表示 453
トレース
オプション 452
プール定義 450
ホストおよびアプリケーション・サーバーの選択 448
ホスト・シミュレーション・ランタイム・トレースを使用可能にする 454
問題判別 450
役割 445
ユーザー・リスト 450
ライセンスの管理 449
ログ
オプションの設定 452
表示 452
HATS の評価 39, 80
HATS プリファレンス
リッチ・クライアント 99
HATS Toolkit 149
HATS マクロ
エクスポート 396
Host On-Demand
カスタム・テーブルのサポート 514
トレース設定 563
Host On-Demand PDT コンパイラー 419
Host On-Demand マクロ
インポート 395
エクスポート 396
Host Publisher
マイグレーション 26
Host Publisher マクロ
インポート 395
エクスポート 396
HTML DDS キーワード・コンポーネント 233
HTTP 圧縮 142
マイグレーション 22

I

IBM Bluemix Server
HATS アプリケーションの開発 49
IBM i 5250 Unicode サポート
双方向 534
IBM License Metric Tool
マイグレーション 22
IBM Tivoli License Compliance Manager
マイグレーション 22
Input Method Editor (IME) 543
DBCS での最大長の除去 544, 545
SBCS での最大長の除去 548
ioPatternKey タグ (runtime.properties) 557
iPad デバイス
考慮事項 62
ITLM
マイグレーション 22

J

J2EE
マイグレーション 30
Java 2 セキュリティー 497
JIS2004
言語サポート 510, 511
JSF Web ページ
統合オブジェクト 411

K

keepOutputTogether 232
Kerberos
使用 496
「セキュリティー」タブ 168

L

logFile tag (runtime.properties) 558
logMask タグ (runtime.properties) 558
LU 名のプロンプト表示
リッチ・クライアント 99

M

maxHODThreadManagerThreads タグ (runtime.properties) 561
maxLogFiles タグ (runtime.properties) 559
maxLogFileSize タグ (runtime.properties) 560
maxTraceFiles タグ (runtime.properties) 560
maxTraceFileSize タグ (runtime.properties) 561

O

OLTP 473
Oracle WebLogic Server
HATS アプリケーションの開発 46
otherParameters タグ 432

P

Page Designer
テンプレート
「ソース」タブ 376
「デザイン」タブ 373
「プレビュー」タブ 376
分割タブ 376
変換
「ソース」タブ 209
「デザイン」タブ 206
「プレビュー」タブ 209
分割タブ 209
Web
テンプレート 372
変換 205
PDT コンパイラー 419

R

recordSimulationTrace タグ (runtime.properties) 565
runtime-debug.properties ファイル 557
リッチ・クライアント・アプリケーション 93
Web アプリケーション 450
runtime.properties タグ
adminPortNum 557
ioPatternKey 557
licenseFile 558
licenseHardLimit 558
licenseTracking 558
logFile 558
logMask 558
maxHODThreadManagerThreads 561
maxLogFiles 559
maxLogFileSize 560
maxTraceFiles 560
maxTraceFileSize 561
numLicenses 557
recordSimulationTrace 565
traceFile 560
traceLogDirectory 561
trace.APPLET 562
trace.HOD.COMMEVENT 563
trace.HOD.DISPLAYTERMINAL 563
trace.HOD.DS 563
trace.HOD.MACRO 563
trace.HOD.OIAEVENT 564

runtime.properties タグ *(続き)*
trace.HOD.PS 564
trace.HOD.PSEVENT 564
trace.HOD.SESSION 564
trace.HOD.TRANSPORT 564
trace.HOD.USERMACRO 564
trace.INTEGRATIONOBJECT 562
trace.RUNTIME 562
trace.RUNTIME.ACTION 562
trace.TRANSFORM 562
trace.TRANSFORM.COMPONENT 562
trace.TRANSFORM.WIDGET 562
trace.UTIL 562
runtime.properties ファイル 557
クラスター環境 43
リッチ・クライアント・アプリケーション 93
Web アプリケーション 450

S

SBCS
最大長の除去 548
screenRecoUseOIASnapshot
接続設定 396
SSH
使用可能化 480
SSL
使用可能化 477
「セキュリティー」タブ 166
マイグレーション 25
EJB プロジェクト 166
SSL 鍵ストア・ファイル
WEL 494
struts Web ページ
統合オブジェクト 410

T

TLCM
マイグレーション 22
Tomcat
HATS アプリケーションの開発 45
traceFile タグ (runtime.properties) 560
traceLogDirectory タグ (runtime.properties) 561
trace.APPLET タグ (runtime.properties) 562
trace.HOD.COMMEVENT タグ (runtime.properties) 563
trace.HOD.DISPLAYTERMINAL タグ (runtime.properties) 563
trace.HOD.DS タグ (runtime.properties) 563
trace.HOD.MACRO タグ (runtime.properties) 563
trace.HOD.OIAEVENT タグ (runtime.properties) 564
trace.HOD.PS タグ (runtime.properties) 564
trace.HOD.PSEVENT タグ (runtime.properties) 564
trace.HOD.SESSION タグ (runtime.properties) 564
trace.HOD.TRANSPORT タグ (runtime.properties) 564
trace.HOD.USERMACRO タグ (runtime.properties) 564
trace.INTEGRATIONOBJECT タグ (runtime.properties) 562
trace.RUNTIME タグ (runtime.properties) 562
trace.RUNTIME.ACTION タグ (runtime.properties) 562
trace.TRANSFORM タグ (runtime.properties) 562
trace.TRANSFORM.COMPONENT タグ (runtime.properties) 562
trace.TRANSFORM.WIDGET タグ (runtime.properties) 562
trace.UTIL タグ (runtime.properties) 562

U

UDC 550
Unicode
サポート 509
URL
に転送 190
表示 189
URL コンポーネント 257
URL ホスト・コンポーネント
マイグレーション 25
UTF-8
エンコードの設定 509

V

Visual Editor
リッチ・クライアント
テンプレート 377
変換 209
VT 表示オプション
双方向 525, 531

W

WAR
ファイル 35
Web アプリケーション
インストール 40
自動切断および最新表示
設定 134
展開 38
Web サービス 403
双方向 517, 530
RESTful サービス 403
Web ブラウザー
複数インスタンス 41
Web プロジェクト
エクスポート
Java EE 39
Web ページの構築
統合オブジェクト 409
WebFacing
HATS 473
HATS 接続 473
HATS 相互運用性 474
WebLogic
HATS アプリケーションの開発 46
WebSphere Portal
HATS 457
WEL
インプリメンテーション 483
計画 483
使用 481
初期化パラメーター 487
「セキュリティー」タブ 168
ネットワーク・セキュリティー・プラグイン 485
マイグレーション 25
ログオン・マクロの作成 484
Credential Mapper の選択 487
Credential Mapper プラグイン 485
SSL 鍵ストア・ファイル 494
WSRP 465

X

xml タグ
otherParameters 432

[特殊文字]

.ear
ファイル 35
.war
ファイル 35

Printed in Japan

SA88-5379-01

日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21