

# IBM i5/OS Program Conversion 使用の手引き (i5/OS V6R1 用)

以前のリリースでの i5/OS V6R1  
プログラム変換の準備

オブジェクト変換分析  
(ANZOBJCVN) ツールの使用

i5/OS V6R1 で実行するための  
プログラムの変換



Mike Charlton  
Theresa Gardiner  
Paul Godtland  
Debbie Landon  
Mayumi Matsukawa  
Ian J. Mills





International Technical Support Organization

**IBM i5/OS Program Conversion 使用の  
手引き (i5/OS V6R1 用)**

2008 年 8 月

**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、v ページの『特記事項』に記載されている情報をお読みください。

本書は i5/OS V6R1 に適用されます。

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: IBM i5/OS Program  
Conversion  
Getting Ready for i5/OS V6R1  
発行: 日本アイ・ビー・エム株式会社  
担当: ナショナル・ランゲージ・サポート

第 1 刷 2008.8

© Copyright International Business Machines Corporation 2008. All rights reserved.

# 目次

特記事項	.v
商標	vi
前書き	vii
この Redpaper を作成したチーム	viii
出版物の著者になりませんか	x
ご意見をお寄せください	x
以前のバージョンの要約	xi
2007 年 7 月のドラフト更新	xi
2007 年 10 月のドラフト更新	xi
2007 年 11 月のドラフト更新	xii
2008 年 1 月のドラフト更新	xiii
2008 年 5 月、この Redpaper の第 1 版	xiii
<b>第 1 章 プログラム変換プロセスの概要</b>	<b>1</b>
<b>第 2 章 プログラム変換の概要</b>	<b>3</b>
2.1 i5/OS の主要な価値	4
2.2 プログラム変換とは何か	4
2.2.1 歴史的展望	6
2.3 プログラム変換がなぜ必要か	6
2.3.1 完全性	6
2.3.2 パフォーマンス	7
2.3.3 機能面の向上	8
<b>第 3 章 変換の準備</b>	<b>11</b>
3.1 オブジェクトの分析	12
3.1.1 用語の定義	12
3.1.2 必要な PTF	13
3.1.3 コマンドの使用上の注意	14
3.1.4 変換時間の見積もり	15
3.2 情報の収集	17
3.2.1 ANZOBJCVN *COLLECT コマンドのパラメーター	18
3.2.2 収集コマンドの例	21
3.2.3 データ収集に使用されるファイル	23
3.3 報告書の作成	23
3.3.1 ANZOBJCVN *REPORT コマンドのパラメーター	25
3.3.2 報告書コマンドの例	26
3.3.3 変換の報告書を理解する	28
3.4 変換されないプログラム	31
3.4.1 失われる属性	33
3.5 分析データに対する QUERY	35
3.6 アプリケーション開発者に関する考慮事項	36
3.6.1 アプリケーションを変換するか	36
3.6.2 その他の決定基準	37
3.6.3 適応コード生成	38
3.6.4 修正とテスト	38
3.7 その他のオブジェクト変換	39

3.7.1 Java プログラム	39
3.7.2 統合ファイル・システムの名前	42
3.7.3 スプール・ファイル	43
3.8 サポートされないプロダクトの削除	44
<b>第 4 章 プログラム変換方式</b>	<b>45</b>
4.1 プログラム変換の概要	46
4.2 STROBJCVN を使用した変換	47
4.2.1 STROBJCVN の使用上の注意	47
4.2.2 STROBJCVN コマンドのパラメーター	48
4.2.3 STROBJCVN OPTION(*CHECK)	51
4.2.4 STROBJCVN OPTION(*CONVERT)	53
4.3 復元時のプログラム・オブジェクトの変換	55
4.4 最初のプログラムの実行時または呼び出し時の変換	57
4.5 変換時間とその改善方法	57
<b>付録 A MRI PTF</b>	<b>59</b>
<b>付録 B ファイルのフィールド定義の一覧表</b>	<b>63</b>
QAIZACVN: 分析されたライブラリーの情報	64
QAIZADIR: 分析されたディレクトリーの情報	67
QAIZAOBJ: 分析された統合ファイル・システム・オブジェクトの情報	68
QAIZASPL: 分析されたスプール・ファイルの情報	70
<b>付録 C 変換を必要とする IBM ソフトウェア・スキップ・シップ・プロダクト</b>	<b>71</b>
<b>関連資料</b>	<b>73</b>
その他の資料	73
オンライン・リソース	73
Redbook の入手方法	73
IBM のサポート	74

# 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。  
〒106-8711 東京都港区六本木 3-2-12 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を提供し、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なります。

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。

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認できません。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。

## 商標

以下は、International Business Machines Corporation の米国およびその他の国における商標です。本書では、これらおよびその他の IBM 商標につき、それぞれが最初に出現する個所で所定のマーク (® または ™) を付けています。現時点での IBM の商標については、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。

AS/400®	Language Environment®	QuickPlace®
Domino®	Lotus Enterprise Integrator®	Redbooks (logo)  ®
DB2®	Lotus®	Redbooks®
eServer™	MQSeries®	Sametime®
Integrated Language Environment®	OS/400®	System i™
IBM®	PartnerWorld®	System i5™
iSeries®	POWER6™	System/38™
i5/OS®	Quickr™	WebSphere®

他の会社名、製品名およびサービス名等はそれぞれ各社の商標です。

Java およびすべての Java 関連の商標およびロゴは Sun Microsystems, Inc. の米国およびその他の国における商標です。



# 前書き

IBM® は、2008 年 3 月に i5/OS® V6R1(IBM i 6.1) の提供を開始しました。V6R1 では、IBM はマシン・インターフェース (MI) アーキテクチャーの機能を活用して、プログラムを大幅に改善しています。プログラムは、パフォーマンスの向上、オペレーティング・システムおよびプロセッサの一連の新機能、および一層強化されたシステム保全性から利益を得ることができます。これらの機能強化を使用可能にするには、以前のリリース用に作成されたすべての MI プログラムを V6R1 で実行できるように変換する必要があります。MI プログラムには、統合言語環境 (ILE) プログラムとオリジナル・プログラム・モデル (OPM) プログラムが含まれます。

プログラムを変換するには、その作成データ (プログラム識別情報のサブセット) が使用可能でなければなりません。MI プログラムはデフォルトで作成データを保存するため、ほとんどのプログラムは、System/38™ 用に最初に作成されたプログラムまでさかのぼって変換できます。外部アクセスからの作成データを除去するオプションを選択した場合でも、プログラムとその構成モジュールのターゲット・リリースが V5R1 以降であれば、ライセンス内部コード (LIC) は作成データへのアクセスが可能です。したがって、プログラム変換用の十分な作成データが得られなくなるのは、プログラムまたは少なくともそのモジュールの 1 つが、IBM OS/400® V4R5 以前のリリース用に作成されたものであり、かつ作成データが明示的に削除された場合に限られます。

V5R4 または V5R3 上でオブジェクト変換分析 (ANZOBJCVN) コマンドを実行すると、i5/OS アップグレードの計画を立てるのに役立ちます。ANZOBJCVN コマンドは、V6R1 で実行するように変換されないプログラムを識別し (ある場合)、プログラム変換に要する時間を見積もります。これは、他の 2 つのタイプの変換 (統合ファイル・システムの名前とスプール・ファイル) に関する情報も提供します。ANZOBJCVN コマンドへのアクセスは、本書に記載されている該当の PTF をロードして適用することにより可能になります。

この IBM Redpaper 資料は、システム管理者および現場技術者が V6R1 へのアップグレードまたはマイグレーションについて理解し、準備するのを支援することを目的としています。本書では、プログラム変換とは何か、なぜ V6R1 上でプログラムを作成または再作成するのかについて説明します。次に、V5R3 または V5R4 上で ANZOBJCVN ツールを使用して V6R1 オブジェクト変換の準備をする手順について詳しく説明します。最後に、V6R1 で実行するようにプログラムを変換するためのプログラム変換方式について説明します。

**i5/OS V6R1 対 IBM i 6.1:** この Redpaper の以前のドラフト版および V6R1 ソフトウェアで使用されるオペレーティング・システム名との整合性を取るために、この Redpaper では、IBM i 6.1 の代わりに i5/OS V6R1 という名称を使用します。

## この Redpaper を作成したチーム

この Redpaper は、International Technical Support Organization (ITSO)、Rochester Center で働く世界各国からの専門家チームによって作成されました。



**Mike Charlton** は、IBM System i™ IT スペシャリストです。以前は高等学校の数学の教師でした。IBM には 1998 年にソフトウェア・サポート・スペシャリストとして参加し、英国の IBM AS/400R サポート・センターに勤務しました。ここでは、ワーク・マネージメント、パフォーマンス、パフォーマンス分析ツールを専門としていました。2002 年にアドバイザー・レベルに昇進しています。現在は、英国の IBM Hursley Laboratories で System i プラットフォーム・スペシャリストを務めています。



**Theresa Gardiner** は、Rochester System i Support Center のスタッフ・ソフトウェア・エンジニアです。1977 年にウィスコンシン州ミルウォーキーの技術員として IBM のキャリアを踏み出しました。1980 年代末にハードウェアからソフトウェア・サポートに方向転換しています。Rochester Support Center に移った後、言語およびユーティリティー・チームでの活動を開始し、現在はそのチーム・リーダーを務めています。Rochester Support Center の勤務時期は、CISC から RISC への変換期に当たっていました。S36E および S38E 言語とユーティリティーのサポート専門家です。



**Paul Godtland** は、ミネソタ州ロチェスターのシニア・ソフトウェア・エンジニアで、System i ライセンス内部コードの設計、実装を担当しています。現在は、MI プログラム変換プロジェクトの技術的指導者です。IBM での 29 年間の大半を、System/38、AS/400、IBM eServer™ iSeries®、System i ファミリー・システム中心に取り組んできました。これまでの経歴では、Integrated Language Environment® (ILE)、AS/400 CISC から RISC への変換、およびテラスペースの設計および実装チームのメンバーも務めています。また、初期の AS/400 C2 セキュリティー格付けの実現や現在進行中のシステム保全性の改善にも貢献しています。



**Debbie Landon** は、IBM ITSO Rochester Center の IBM 認定シニア IT スペシャリストです。現在の専門分野は、Sametime® や QuickPlace® のような Domino® および関連 Lotus® 製品を含む、System i コラボレーション製品が中心です。IBM には 23 年間勤務しており、最初は S/36 を担当し、次に担当した AS/400 が System i プラットフォームへと発展しています。2000 年 11 月に ITSO に参加する前は、IBM Partne® WorldR for Developers iSeries チームのメンバーとして、Domino for iSeries の分野で IBM ビジネス・パートナーをサポートしていました。



**Mayumi Matsukawa** は、日本の System i テクニカル・サポート・チームの IT スペシャリストです。1998 年に IBM に参加し、System i の作業に関して 8 年の経験を積んでいます。専門分野には、Domino と関連 Lotus 製品、LPA®、クラスタリング、サイト間ミラーリング (XSM)、DB2R for i5/OS が含まれます。IBM 専門家、ビジネス・パートナー、およびお客様を対象に、技術サポートと教育を行っています。



**Ian J. Mills** は、英国の iSeries サポート・センターのアドバイザー IT ソフトウェア・サポート・スペシャリストです。1995 年にソフトウェア・サポート・スペシャリストとして大学から直接 IBM に参加し、2001 年にアドバイザー・レベルに昇格しました。サポート関係の仕事だけでなく、200 を超えるシステム・アップグレード契約に直接携わり、英国とアイルランド全域のお客様のために数多くのシステム・リカバリーを手掛けています。2007 年 3 月に、i5/OS スペシャリストとして再認定され、2007 年 7 月以降、英国で IBM アベイラビリティ・マネージャーとして働いています。ポーツマス大学を卒業し、IT の学位を取得しました。

本プロジェクトに対する以下の方々のご支援に感謝いたします。

Margaret Fenlon  
Kary Garvey  
Ian Jarman  
Rodney Johnson  
Laurie Miller  
Gerri Passe  
Rita Roque  
Gottfried Schimunek  
Michelle Schlicht  
Tom Severson  
Judy Trousdell  
Kevin Vette  
IBM Rochester 開発チーム

Rodolfo Segoviano Rayas  
IBM Guadalajara, Mexico

Al Barsa, Jr.  
Barsa Consulting Group, LLC

Thomas Gray  
Joanna Pohl-Misczyk  
Craig Schmitz  
Jenifer Servais  
ITSO, Rochester Center

## 出版物の著者になりませんか

2 週間から 6 週間の研修プログラムにご参加いただけます。最先端のテクノロジーを使用して実践経験を積みながら、特定の製品やソリューションに関する資料の作成にご協力ください。IBM の専門技術者、ビジネス・パートナー、お客様とチームを組む機会が得られます。

貢献いただくことで、製品の認知度やお客様の満足度をいっそう高めることができます。さらに、IBM 開発研究所との人脈を築き、生産性や市場性を高めるのにも役立ちます。

研修プログラムについての詳細は、次の Web サイトで Residency Index を参照してください。参加のお申し込みは、オンラインで受け付けています。

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## ご意見をお寄せください

読者の皆様からのご意見をお待ちしております。

できるだけ皆様のお役に立つ Redpaper を提供したいと考えております。この Redpaper や他の IBM RedbookR に関するご意見を、以下のいずれかの方法でお送りください。

- ▶ 以下のサイトからオンラインの「**Contact us**」の Redbook ご意見フォームを使用する

[ibm.com/redbooks](http://ibm.com/redbooks)

- ▶ 以下のアドレス宛てに電子メールを送信する

[redbooks@us.ibm.com](mailto:redbooks@us.ibm.com)

- ▶ 以下の宛先にコメントを郵送する

IBM Corporation, International Technical Support Organization  
Dept. HYTD Mail Station P099  
2455 South Road  
Poughkeepsie, NY 12601-5400

# 以前のバージョンの要約

この Redpaper は、この第 1 版のリリースに先だって、2007 年 7 月の i5/OS V6R1 のプレビューと同時にドラフト形式で提供されるようになりました。このセクションでは、本書の以前のバージョンで行われた一連の技術的変更について説明します。小さい修正や編集上の変更については取り上げません。

## 2007 年 7 月のドラフト更新

この Redpaper の第 1 ドラフトには、以下の情報が含まれています。

- ▶ 第 1 章、『プログラム変換の概要』
- ▶ 第 2 章、『変換の準備』
- ▶ 第 3 章、『プログラム変換方式』

この章には、変換プロセスまたは変換方式に関する最小限の情報のみが含まれています。

- ▶ 付録 A、『MRI PTF』

オブジェクト変換分析 (ANZOBJCVN) CL コマンドは、V5R3 用および V5R4 用の PTF として配布されています。この Redpaper のこのドラフト版には、V5R3 用 (SI27515) と V5R4 用 (SI25550) の英語の PTF の *オリジナル・バージョン* の PTF 番号のみが含まれています。

- ▶ 付録 B、『プログラム変換の要約』
- ▶ 付録 C、『ファイル・フィールド定義』

## 2007 年 10 月のドラフト更新

この Redpaper の第 2 ドラフトには、以下の情報が含まれています。

- ▶ 第 2 章、『変換の準備』

- ANZOBJCVN \*REPORT コマンドの新規 \*CVNPRB パラメーターに関する情報を追加しました。この新しい報告書パラメーターは、直前のライブラリー収集で指定された各ライブラリーごとに、変換されないプログラム・オブジェクトの詳細報告書を提供します。

- ANZOBJCVN \*REPORT コマンドの新規 \*ALLAVL パラメーターの説明を追加しました。このパラメーターは、以前のデフォルトの \*ALL パラメーターに取って代わるものです。\*ALLAVL パラメーターを指定すると、収集された情報のタイプごとに報告書が生成されます。

- ▶ 付録 A、『MRI PTF』

オブジェクト変換分析 (ANZOBJCVN) CL コマンドは、V5R3 用および V5R4 用の PTF として配布されています。この Redpaper のこのドラフト版には、V5R3 用 (SI28425) と V5R4 用 (SI28415) の英語の PTF の *更新バージョン* の PTF 番号のみが含まれています。

## 2007 年 11 月のドラフト更新

この Redpaper の第 3 ドラフトには、以下の情報が含まれています。

### ▶ 第 3 章、『変換の準備』

アップグレード前に V6R1 上のサポートされないプロダクトを除去することを推奨する記載を追加しました。

この章には、以下の変更が含まれています。

#### – 3.2.1、『ANZOBJCVN \*COLLECT コマンドのパラメーター』

- QSYS2 ライブラリーに関する情報が追加され、何も処置を取る必要がないことが記載されています。
- サブツリー・パラメーターの説明が拡張されています。

#### – 3.2.2、『収集コマンドの例』

\*CVNPRB パラメーターに関する情報が追加され、関連画面のキャプチャーが更新されました。

#### – 3.4、『変換されないプログラム』

追加情報が追加されました。

#### – 3.5、『分析データに対する QUERY』

コマンドのファイルが別のライブラリーに移動された場合、QUERY は ANZOBJCVN の結果をどのように報告する必要があるかについての情報が追加されました。

#### – 3.6、『アプリケーション開発者に関する考慮事項』

適応コード生成 (ACG) に関する情報が追加されました。

#### – 3.7.3、『スプール・ファイル』

このセクションは更新されて、アップグレード前にスプール・ファイルのハウスキューピングを実行するようとの助言が組み込まれています。

### ▶ 第 4 章、『プログラム変換方式』

この章には、以下の更新が含まれています。

#### – 4.1、『プログラム変換の概要』(新しいセクションの追加)

#### – 4.2、『STROBJCVN を使用した変換』(内容の改訂)

マルチスレッド化に関する情報が 4.2.1、『STROBJCVN の使用上の注意』に追加されました。

#### – 4.3、『復元時のプログラム・オブジェクトの変換』(内容の改訂)

#### – 4.5、『変換時間とその改善方法』(新しいセクションの追加)

### ▶ 付録 A、『MRI PTF』

オブジェクト変換分析 (ANZOBJCVN) CL コマンドは、V5R3 用および V5R4 用の PTF として配布されています。この Redpaper のこのドラフト版では、各国語の PTF 番号が提供されています。

### ▶ 付録 B、『プログラム変換の要約』

この付録は変更され、クイック・リファレンス・ガイドが追加されています。

### ▶ 付録 D、『変換を必要とする IBM ソフトウェア・スキップ・シップ・プロダクト』

この新しい付録には、V6R1 へのアップグレード時に変換する必要があるスキップ・シップ IBM プログラム・プロダクトのリストが掲載されています。

## 2008 年 1 月のドラフト更新

この Redpaper の第 4 ドラフトには、以下の情報が含まれています。

- ▶ vii ページの『前書き』  
前書きと要約が更新され、新しい序文が含まれています。
- ▶ 1 ページの第 1 章、『プログラム変換プロセスの概要』  
新しい章を設けて、プロセスの簡単な要約を示し、プログラム変換の計画と実装に役立つようにしています。この情報は、この Redpaper の以前のバージョンでは付録に含まれていたものです。

## 2008 年 5 月、この Redpaper の第 1 版

この Redpaper の各ドラフト版で段階的に進められてきた変更に加えて、この第 1 版には以下の情報が含まれています。

- ▶ 13 ページの 3.1.2、『必要な PTF』  
ANZOBJCVN コマンドは、このセクションにリストされている新しいコード PTF が提供されるようになりました。主な変更点は、変換時間の見積もりの際に、1 つの完全なプロセッサより小さいプロセッサが割り当てられた区画を考慮するようになったことです。
- ▶ 15 ページの 3.1.4、『変換時間の見積もり』  
このセクションには、ANZOBJCVN コマンドを実行するシステムとは異なるシステムを見積もる際の部分プロセッサの指定方法に関する情報が含まれています。
- ▶ 45 ページの第 4 章、『プログラム変換方式』  
この章は、3 つの変換方式のそれぞれで可能な並行処理に関する説明が追加されています。さらに、V6R1 システム上でいかなる種類の変換も開始する前に、必ず PTF MF44237 をインストールしておく必要があるという情報が追加されました。この PTF は、プログラム変換中にまれに発生するシステム・ハングを防止します。







## プログラム変換プロセスの概要

この章では、プログラム変換を計画し、実装するためのプロセスを要約します。この章は、V6R1 プログラム変換プロセスに必要なステップの簡易ガイドとしてご利用いただけます。この概要の中で強調表示されているステップは、本書内のさらに詳しい情報が記載されている別の章への参照を含んでいます。本書全体をお読みになり、V6R1 に必要なプログラム変換について完全に理解することをお勧めします。

**注：**プログラム変換の計画は V5R3 または V5R4 システムで開始し、実際の変換は V6R1 へのアップグレード後に行われます。

以下のステップは、プログラム変換を計画し、実装するためのプロセスを簡単に要約しています。

## ステップ 1: V5R4 または V5R3 PTF を適用して ANZOBJCVN コマンドを使用可能にする

大抵の場合、希望する各国語の MRI PTF を発注すると、必要なコード PTF も自動的に出荷されます。13 ページの 3.1.2、『必要な PTF』を参照してください。

## ステップ 2: システム上のオブジェクトに関する情報を収集する

ANZOBJCVN OPTION(\*COLLECT) コマンドを実行します。指定されたタイプを収集するたびに、そのタイプの収集ファイル (複数の場合あり) 内のデータを置き換えます。そのため、17 ページの 3.2、『情報の収集』に記載されている情報、特に 23 ページの 3.2.3、『データ収集に使用されるファイル』で提供される情報は慎重に考慮してください。

## ステップ 3: 収集された情報を検討し、理解する

収集されたデータに対して、ANZOBJCVN OPTION(\*REPORT) コマンドを実行し、オプションとしてユーザー独自の QUERY を実行します。以下の報告書を考慮してください。

- ▶ \*CVNPRB パラメーターを使用した、変換しないプログラムに関する報告書
- ▶ \*LIBSUM パラメーターを使用した、要約報告書情報 (各ライブラリーの推定変換時間など)

23 ページの 3.3、『報告書の作成』および、特に 28 ページの 3.3.3、『変換の報告書を理解する』を参照してください。

## ステップ 4: アプリケーション提供者と連絡を取る (必要な場合)

購入済みのアプリケーションが変換されないことが分かり、それを再コンパイルするためのソース・コードを持っていない場合は、アプリケーション提供者と連絡を取り、V6R1 上で実行できるアプリケーションのバージョンを入手します。31 ページの 3.4、『変換されないプログラム』を参照してください。

## ステップ 5: サポートされないプロダクトを除去してから、V6R1 にアップグレードする

V6R1 にマイグレーションする前に、サポートされないライセンス・プログラム・プロダクト (LPP) およびその他のサポートされないプロダクトを除去します。44 ページの 3.8、『サポートされないプロダクトの削除』を参照してください。

オブジェクト変換のための準備作業がすべて完了した後、以下の Web アドレスで、「i5/OS および関連ソフトウェアのインストール、アップグレードおよび削除」(SD88-5002-10) を参照してください。

<http://publib.boulder.ibm.com/infocenter/systems/topic/rzahc/sc415120.pdf>

## ステップ 6: V6R1 上でオブジェクトを変換する

ユーザーの環境に最も適した変換オプションの組み合わせを選択します。以下のセクションを参照してください。

- ▶ 47 ページの 4.2、『STROBJCVN を使用した変換』
- ▶ 55 ページの 4.3、『復元時のプログラム・オブジェクトの変換』
- ▶ 57 ページの 4.4、『最初のプログラムの実行時または呼び出し時の変換』



## プログラム変換の概要

プログラム変換は、i5/OS V6R1 にアップグレードするすべてのシステムに必要です。本書の情報と、i5/OS V5R3 および V5R4 用に提供されたツールを使用して、この変換のための準備をしてください。i5/OS V6R1 へのアップグレードを実施する前に十分に準備することによって、移行を円滑に進めることができます。

**概要：** V6R1 へのアップグレードの準備方法についての簡単な概要は、1 ページの第 1 章、『プログラム変換プロセスの概要』を参照してください。

## 2.1 i5/OS の主要な価値

i5/OS 稼働環境は、リレーショナル・データベース、セキュリティー、Web サービス、ネットワーク、およびストレージ管理機能を、信頼できる組み合わせで統合しています。この統合により操作の単純化が推進され、特に中小規模ビジネスのお客様に高く評価されています。また i5/OS は、ビジネス・プロセス・アプリケーションを効率的にデプロイするための幅広い、非常に安定したミドルウェア基盤を備え、2,500 社を超える ISV の 5,200 を超えるソリューションをサポートしています。i5/OS ソリューションは、幅広い高度のスキルを持つ IBM ビジネス・パートナーのネットワークを通して全世界に提供され、IBM の信頼できるサービスおよびサポート・インフラストラクチャーに支えられています。

i5/OS は、非常にスケーラブルでウィルス耐性の高いアーキテクチャーを備え、卓越したビジネス回復力が立証されています。大規模企業の場合、ビジネス・プロセスの回復力と、複数のビジネス単位で要求される厳しいサービス・レベルを満たし、かつそれを超える能力を身に付けることが、最優先課題となっています。i5/OS は、長年にわたり、さまざまなビジネス分野で、回復力のあるアプリケーション・デプロイメントを実現する業界トップレベルの稼働環境の 1 つとして、その評価を確立しています。企業は日々 i5/OS 稼働環境を信頼して、企業の最重要ビジネス・アプリケーションをデプロイしています。

i5/OS は、企業がその IT 投資から最大限の収益が得られるように、多様な拡張オプションを提供しています。稼働を中断せずに拡張するために最も重要な機能は、i5/OS のテクノロジーから独立したマシン・インターフェース (MI) です。長年にわたり何世代ものテクノロジーを経て実証済みの i5/OS MI は、ハードウェア装置の変更やプロセッサ世代の変更からアプリケーションを守るだけでなく、アプリケーションを再コンパイルせずに新規テクノロジーにアップグレードすることも可能にしています。

## 2.2 プログラム変換とは何か

V6R1 では、システム設計とプロセッサ機能の大幅な改善を活用するために、MI を使用するすべてのアプリケーションを実行前に再作成する必要があります。このような MI プログラムには、統合言語環境 (ILE) プログラムとオリジナル・プログラム・モデル (OPM) プログラムが含まれます。変換により既存のプログラム・オブジェクトは置き換えられますが、それぞれのプログラム・オブジェクトは、名前、ライブラリー、所有するユーザー・プロファイルなどの属性を保持します。この変換は、各オブジェクトに対する一回限りのプロセスです。十分な作成データを持っている MI プログラムは、作成されたリリースに関係なく、すべて変換できます。実際に、当初 System/38 用に作成されたプログラムでさえも変換可能です。

i5/OS Portable Application Solutions Environment (PASE) および Java™ プログラムは、MI に基づいていません。したがって、これらは直接的な影響を受けません。ただし、ディレクトリー内の Java プログラムについても単純な変換が行われます。詳しくは、39 ページの 3.7.1、『Java プログラム』を参照してください。

プログラムは V6R1 で新しいフォーマットを使用します。そのため、以前のリリース用に作成されたプログラムは、V6R1 (または、それ以降のリリース) 上で、実行する前に変換する必要があります。同様に、V6R1 またはそれ以降のリリースで作成されたプログラムは、V5R4 またはそれ以前のリリース上で、実行する前に変換が必要です。

System i プラットフォームによって使用されるソフトウェア設計およびハードウェアは、完全性、パフォーマンス、機能性などのシステムの重要な側面で、大幅な改善を実現できる時点に達しています。これは、システム・アーキテクチャーを利用してプログラムを変換することにより実現されます。詳しくは、6 ページの 2.3、『プログラム変換がなぜ必要か』を参照してください。

プログラムの変換 (または、トランスフォーメーション) は、3 つの異なる方法で実行できます。

- ▶ プログラムの復元時またはアプリケーションのインストール時 (アプリケーションが V6R1 用に特別に生成されたものでない場合)
- ▶ 明示的にスケジュールされた時点
- ▶ プログラムが最初に実行される時点

変換方式を自由に組み合わせて、ユーザー独自の状態に適合させることができます。変換方式について詳しくは、45 ページの第 4 章、『プログラム変換方式』を参照してください。

お客様がプログラム変換の準備をするのを支援するために、オブジェクト変換分析 (ANZOBJCVN) コマンドが、V5R3 および V5R4 システムに適用できる PTF の形で提供されています (この 2 つのリリースは、直接 V6R1 にアップグレードできます)。お客様は ANZOBJCVN コマンドを使用してシステムを分析し、V6R1 に変換されないプログラム・オブジェクトがあるかどうかを調べることができます。また、このコマンドは、プログラム変換にかかる時間の見積もりも提供します。

V6R1 にアップグレードする前に、ANZOBJCVN CL コマンドを使用することをお勧めします。既存のシステム上で V6R1 にアップグレードする前に、あるいは V6R1 を実行する新規ハードウェアに移行する前に、このツールを使用することにより、適切な計画を完了させて、必要な事前対策を講じることができます。プログラムが変換されない場合は、それを再コンパイルするか、あるいは新しいバージョンのプログラムをソフトウェア提供者から入手する必要があります。

ANZOBJCVN コマンドは、V6R1 にアップグレードまたはマイグレーションする予定のすべての V5R4 および V5R3 i5/OS 論理区画 (LPAR) に対して使用してください。また、独立した補助記憶域プール (ASP) には、変換を必要とするオブジェクトが存在する可能性があるため、計画の段階でこれを対象に含めることも考慮する必要があります。ANZOBJCVN コマンドの使用についての詳細は、11 ページの第 3 章、『変換の準備』を参照してください。

オブジェクト変換は、以前のターゲット・リリースを指定して V6R1 上で作成されたプログラムを V5R4 または V5R3 システムに復元する際にも実行されます。V6R1 プログラムの復元先の旧リリース上に必要な PTF については、13 ページの 3.1.2、『必要な PTF』を参照してください。

図 2-1 は、どのような場合にプログラム変換が行われるか、または必要であるかを示しています。i5/OS V6R1 が IBM POWER6™ プロセッサを活用する方法についての詳細は、7 ページの 2.3.2、『パフォーマンス』、および 38 ページの 3.6.3、『適応コード生成』を参照してください。

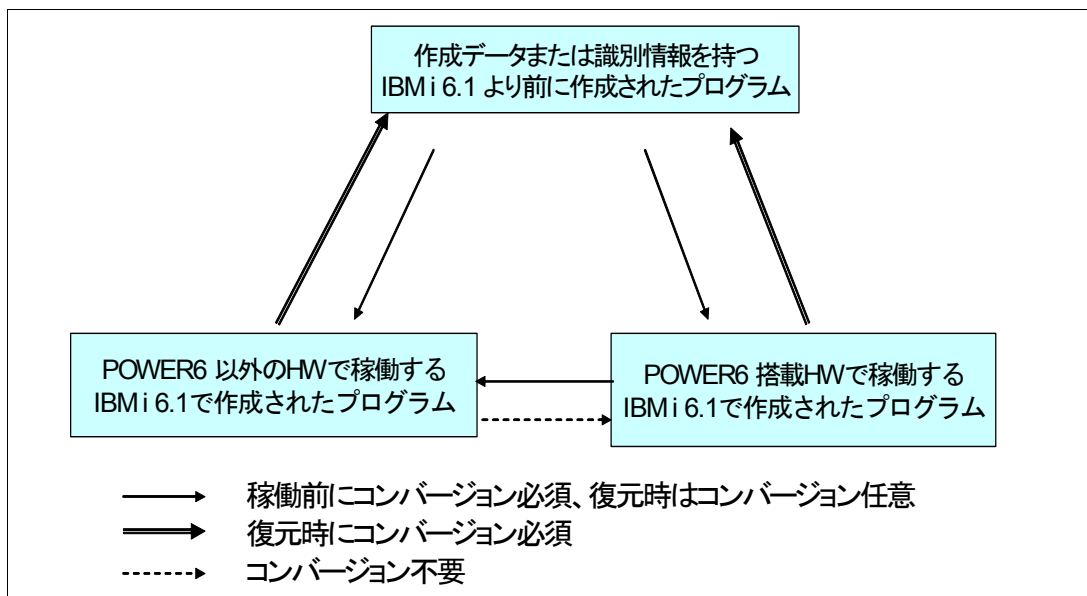


図 2-1 プログラム変換図

## 2.2.1 歴史的展望

今回の変換は、MI アーキテクチャーの歴史の中で 3 度目になります。最初の変換は、1988 年に System/38 から OS/400 V1R1 に移行したときに実施されました。2 回目の変換は、1995 年に OS/400 V3R6 に対して行われました。このときは、48 ビット・アドレスから 64 ビット・アドレスにアップグレードされ、異なるハードウェア命令セットが実行された時期です。今回の変換は過去の変換に比べると単純です。例えば、プログラム以外のタイプのオブジェクトのほとんどは今回は変更されません。プログラム変換も今回は、はるかに高速で実行されます。

## 2.3 プログラム変換がなぜ必要か

プログラムを V6R1 上で作成または再作成する理由は、これによりシステムのいくつかの重要な側面を大幅に機能強化できることにあります。機能強化には、完全性、パフォーマンス、および機能が含まれます。既存のプログラムが変換され、新しく作成されたプログラムと同じ利益が得られます。

こうした目に見える効果に加えて、プログラム変換は、MI アーキテクチャーの技術的な活力を柔軟にするための 1 つの方法でもあります。MI アーキテクチャーは新しいものではありませんが、業界有数のリフレッシュ能力と改善能力を備えています。すべての MI プログラムをリフレッシュする一方で、その作成データを保持するという組み合わせにより、System i ユーザーは、将来的なハードウェア、システム設計、および業界全体の変更に非常に柔軟に対応できるようになります。

### 2.3.1 完全性

システムが意図された機能または設計された機能のみを正常に実行する場合、そのシステムは完全性があると言えます。すべての MI プログラムを再作成することによる即座の効果は、サポートされない変更（例えば、サービス・ツールでハードウェア命令ストリームを変更すること）が根絶されることです。これらは除去されてしまうため、システムの意図した使用を妨害する機会がありません。

また、将来的なコード破損につながる可能性を除去するシステム固有の機能も拡張されています。システム上で実行できる MI ベースのアプリケーションはすべて、いつでもリフレッシュできるようになりました。例えば、あまりよく知らないベンダーから購入するアプリケーションに心配がある場合、そのプログラムのインストール時に変換することができます。

変換はプログラムをその MI アーキテクチャー構造から再作成するので、システムの完全性が攻撃されることはありません。変換後のプログラムは MI セマンティクスに準拠しており、サポートされるオブジェクト・タイプに対する定義済み操作だけが許可されます。また、他の多くのシステムとは異なり、誰かが新しい形のアタック・プログラムを送りつける前にウイルス・スキャン・ソフトウェアを更新するという、はてしない競争に頭を悩ませる必要もありません。ハードウェア命令のどのシーケンスが問題を起す可能性があるかを心配する代わりに、より高いレベルの MI 操作からプログラムをリフレッシュすることにより、適切なハードウェア命令をシステムに生成させることができます。

もちろん、MI は通常、リリースごとに拡張されますが、こうした変更は、首尾一貫した全体の一部として組み込まれます。各システム・リリースで改善が行われますが、最後に出荷されたリリース以降に発生するアタックから安全に守るための更新の必要性はありません。

V6R1 の完全性の改善でもう 1 つ重要なことは、システム状態のプログラムをロードするには i5/OS の一部にする必要があることです。以前のリリースでは、システム状態のプログラムは、オペレーティング・システムの一部であるかどうかに関係なく、オプションでロードが許可されました。システム状態は、オペレーティング・システムで使用するよう設計されたものです。そのため、過去には、お客様が不注意にオペレーティング・システムの完全性を損なう可能性がありました。システム状態で実行されたプログラムが、オペレーティング・システムと正常に相互運用するように設計されていなかったために、システム破壊のような重大な問題を引き起こすという状態が生じていました。これは、謂わば、こうした他のプログラムがオペレーティング・システム資源の i5/OS 銀行口座から預金を引き出し、結局は残高がマイナスになるようなものでした。

## 2.3.2 パフォーマンス

パフォーマンスの向上は、プログラム変換のもう 1 つの重要な目的です。頻繁に実行される操作が改善され、パフォーマンス上の落とし穴が除去されました。また、プログラム作成時の新しいオプションにより、パフォーマンスの 2 つの側面に重点を置いた処理が追加され、将来的なパフォーマンスの向上が可能になりました。

活動化 (プログラムを実行できるように準備すること)、プロシージャー (関数) 呼び出し、およびポインターの使用が、すべて高速化されています。改善の度合いは、これらの操作が各アプリケーションの全体的パフォーマンスに与える影響の程度によって異なります。

### テラスペース

テラスペースは、各ジョブにローカルの大きな連続したアドレス可能域です。このテラスペースが、ソフトウェア・エミュレーションの代わりに、ハードウェアの直接サポートにより実装されるようになりました。以前のリリースで見られた 1 つの重大な例として、ISV アプリケーションがソフトウェア・エミュレーションの影響を受け、アプリケーションの初期パフォーマンスが全体で 6 倍も低下することがありました。この問題が確認されたときは、ライセンス内部コード・オプション (LICOPT) を使用してアプリケーションを再作成することで、ほとんどのオーバーヘッドが解消されました。テラスペースがハードウェアで直接サポートされるようになった現在は、この潜在的なパフォーマンス上の落とし穴は除去されており、プログラム作成時に特別な考慮をする必要はありません。

エミュレーションによるオーバーヘッドの影響を受けないプログラムでも、ハードウェアを直接使用することにより全般的な利益が得られます。例えば、一般的に使用されるメモリー

重視の i5/OS PASE アプリケーションでは、15% を超えるパフォーマンスの向上が得られません。i5/OS PASE プログラムは変換されないにもかかわらずこのようなことが起きるのは、i5/OS プログラムと MI プログラムがテラスペースを使用して相互作用するためです。メモリー処理効率の全般的な改善も、ILE ユーザーにとってのテラスペースの利点です。

## プログラム作成時の新規オプション

V6R1 をターゲットとするプログラムのサービス・プログラム作成 (CRTSRVPGM) CL コマンドとプログラム作成 (CRTPGM) CL コマンドに、2 つの新しい作成時オプションが提供されました。どちらのオプションもソース・コードの変更は必要ありません。ARGOPT パラメーターは、このオプションを使用して作成されたサービス・プログラムまたはプログラム内のプロシージャ間のすべての呼び出しを最適化します。呼び出しを多用するアプリケーションでは、全体で最大 20% のパフォーマンス向上が得られています。

サービス・プログラムは、BNDSRVPGM パラメーターで \*DEFER 修飾子を使用することにより、活動化を据え置くことも可能になりました。据え置きサービス・プログラム活動化は、これらのサービス・プログラムにバインドするプログラムの活動化時ではなく、必要な時点でのみ活動化されます。これにより、まれにしかメインライン処理で使われない機能のために、プログラム開始時にコストを払わないことを選択できます。

## 適応コード生成

適応コード生成は、特に最新のハードウェアを購入するお客様にとって興味深い、V6R1 のもう 1 つのパフォーマンス向上です。従来は、特定のオペレーティング・システム・リリース用に作成されたプログラムに生成されるコードは、そのリリースに共通の (最小公倍数の) ハードウェア機能だけを使用できました。これにより、そのリリースでサポートされる任意のハードウェア上でプログラムを実行できましたが、このことは、一部の新しいハードウェア機能はいくつかのリリースで活用できないことを意味していました。

これに対して、現在の MI プログラムは、デフォルトで、そのプログラムが作成されたハードウェアで使用可能な機能を使用するようになりました。例えば、V6R1 以降のリリースをターゲットとする、POWER6 プロセッサ・ベースのシステム上で作成された MI プログラムには、POWER6 で初めて使用可能になったハードウェア命令やその他のハードウェア機能が組み込まれます。このようなプログラムが、より古いプロセッサ・ベースのシステムに移動された場合、プログラムは自動的に変換されて、サポートされるすべてのプロセッサ上で使用可能な共通のハードウェア機能のみを使用するようになります。

新しい作成時オプションおよび適応コード生成機能について詳しくは、「*System i ILE 概念バージョン 6 リリース 1*」(SD88-5033-08) を参照してください。

<http://publib.boulder.ibm.com/infocenter/systems/topic/books/sc415606.pdf>

38 ページの 3.6.3、『適応コード生成』も参照してください。

### 2.3.3 機能面の向上

V6R1 の機能面の向上は、アプリケーションの開発とデプロイを容易にすることに重点が置かれています。おそらく、最大の変更点は、すべての MI プログラムがテラスペースを使用できるようになったことです。以前は、テラスペースに対応していないプログラムにテラスペース・アドレスを渡さないように注意する必要がありました。この制約のために、大きな連続アドレス可能域を使用するとより効率的になるケースや、テラスペースへの 8 バイト・タグなしポインターを使用すると C や C++ プログラムのパフォーマンスが向上するような場合に、テラスペースの利点を生かすににくかった面があります。現在は、すべての MI プログラムがテラスペース・アドレスを扱うことができるので、テラスペースの使用がはるかに容易になっています。



その他の機能面の向上としては、各ジョブが約 100 倍の大きさのテラスペースを割り振ることが可能になったことがあり、この割り振りは厳密にジョブにローカルにすることができません。過去には、一部の条件下では、これより小さい割り振りが不注意に複数のジョブ間で参照される可能性がありました。例えば、この本来はジョブにローカルの資源を誤ってジョブ間で使用しようとしていることがテスト時に検出されず、お客様に提供されるアプリケーションに潜在的な問題を残してしまうようなケースがありました。

もう一つの機能強化は、MI プログラムでスレッド・ローカルな静的ストレージが使えるようになったことです。このストレージ・クラスの追加により、スレッド対応のプログラミングがより単純になります。RPG、C++、および C 言語はすべて、このストレージ・クラスにアクセスするための新しい構文を提供しています。RPG は、H 仕様書で「THREAD(\*CONCURRENT)」を使用して、プログラム全体に対してスレッド・ローカルの使用を指定します。代わりに、D 仕様書に「static(\*allthread)」修飾子を追加して、RPG の個々の変数でスレッド・ローカル・ストレージの使用を指定することもできます。C と C++ は、個々の変数宣言で「\_\_thread」修飾子を指定します。詳細については、以下の Web アドレスで、「System i ILE 概念 バージョン 6 リリース 1」(SD88-5033) 資料を参照してください。

<http://publib.boulder.ibm.com/infocenter/systems/topic/books/sc415606.pdf>

Performance Explorer (PEX) およびトレース機能の適用範囲の拡張により、アプリケーションの開発も容易になりました。これらのシステム機能を使用する前に、特殊なオプションを使用して、プログラムを再コンパイルする必要がなくなりました。





## 変換の準備

**重要：**この章で説明するプロセスは、i5/OS を V6R1 にアップグレードする前に使用するものです。分析プロセスの使用は必須ではありませんが、実行することをお勧めします。

この章では、V6R1 オブジェクト変換を計画するプロセスについて説明します。変換を必要とするオブジェクト・タイプは、ライブラリー内のプログラム、統合ファイル・システム内の Java プログラム、統合ファイル・システム内の一部のファイルの名前、およびスプール・ファイルです。変換は V6R1 まで実行されませんが、変換の計画は V5R3 上または V5R4 上で完了させておく必要があります。変換計画は、V6R1 にアップグレードまたはマイグレーションする予定のすべてのシステムおよび区画で完了させる必要があります。

変換の準備をする際には、本書のガイドラインと以下の資料で提供される情報を併せてご使用ください。

- ▶ *i5/OS および関連ソフトウェアのインストール、アップグレードおよび削除* (SD88-5002-10)  
<http://publib.boulder.ibm.com/infocenter/systems/topic/rzahc/sc415120.pdf>
- ▶ System i i5/OS プログラム資料説明書 バージョン 6 リリース 1  
<http://publib.boulder.ibm.com/infocenter/systems/scope/i5os/topic/rzaq9/rzaq9.pdf>
- ▶ System i アップグレード計画 Web ページ：  
<http://www-304.ibm.com/systems/support/i/planning/upgrade/index.html>

## 3.1 オブジェクトの分析

i5/OS バージョン V5R3 および V5R4 用に、オブジェクト変換分析 (ANZOBJCVN) CL コマンドを使用するための PTF が提供されています。お客様はこのツールを使用してオブジェクトを分析し、その分析に基づいて報告書を生成することができます。報告書は、コマンドを実行するシステム上で、スプール・ファイルとして入手できます。

ANZOBJCVN コマンドは、変換に伴う潜在的な障害があるかどうかを識別し、プログラム変換に必要な時間を見積もるのに役立つツールです。これは、他の 2 つのタイプの変換 (統合ファイル・システムの名前とスプール・ファイル) に関する情報も提供します。

ANZOBJCVN コマンドでは、ライブラリー内の以下のタイプのオブジェクトを分析し、報告することができます。

- ▶ プログラム (\*PGM)、具体的には、オリジナル・プログラム・モデル (OPM) プログラムと統合化言語環境 (ILE) プログラム
- ▶ サービス・プログラム (\*SRVPGM)
- ▶ モジュール (\*MODULE)
- ▶ SQL パッケージ (\*SQLPKG)

ほとんどの場合、ANZOBJCVN コマンドで分析されるオブジェクトは、V6R1 上で使用するために変換可能です。変換プロセスについての詳細は、45 ページの第 4 章、『プログラム変換方式』を参照してください。

ディレクトリー内の以下のタイプのオブジェクトも ANZOBJCVN コマンドで分析し、報告することができます。

- ▶ i5/OS オペレーティング・システム上で実行するように最適化された Java プログラムを含むストリーム・ファイル。  
「ルート」 (/)、QOpenSys、およびユーザー定義ファイル・システム内のストリーム・ファイルのみが分析されます。詳しくは、39 ページの 3.7.1、『Java プログラム』を参照してください。
- ▶ 「ルート」 (/) ファイル・システム内およびオプション CASE(\*MONO) を指定して作成されたユーザー定義ファイル・システム内の統合ファイル・システムのオブジェクト名。  
詳しくは、42 ページの 3.7.2、『統合ファイル・システムの名前』を参照してください。

スプール・ファイル・データも ANZOBJCVN コマンドで収集され、システム・アップグレードを計画する際に役立つ情報を提供することがあります。デフォルトでは、スプール・ファイルの変換は、システムの IPL 時、または独立補助記憶域プール (ASP) をオンに構成変更する操作時に実行されます。スプール・ファイルの変換についての詳細は、43 ページの 3.7.3、『スプール・ファイル』を参照してください。

### 3.1.1 用語の定義

以下の用語は、ANZOBJCVN コマンドで生成される報告書に使用され、十分に理解しておくことが必要です。

#### 状態

マシン・インターフェース (MI) プログラムを実行する方法を制御する属性。ユーザー状態またはシステム状態のいずれかです。この状態属性は、プログラムがハイ・セキュリティー・レベルのシステム・ドメイン・オブジェクトに対するアクセス権限を持つかどうかを制御します。さらに、状態属性は、プログラムのユーザー・ドメイン・スペースに対するアクセス権限のタイプ (アクセス権限なし、読み取り専用、または読み取り / 書き込み) も決定します。アプリケーション

ン・プログラムはユーザー状態で実行されます。一方、ほとんどのオペレーティング・システム・プログラムはシステム状態で実行されます。

#### デジタル署名

オブジェクトが変更されていないこと、およびオブジェクトが予期されたソースから提供されたことを確認するために使用される検証。ほとんどの場合、デジタル署名は、オブジェクトがシステムに復元される前のオブジェクトの完全性を検査するために使用されます。デジタル署名は、プログラムなどいくつかのタイプのオブジェクトに、オプションで適用できます。これはプログラム変換時に除去されてしまいます。変換時にはプログラムが変更され、以前のデジタル署名は無効になるためです。

#### プロファイル作成

プログラムのパフォーマンスをさらに向上させるための、データ固有の情報を使用した高度なアプリケーションの最適化。プロファイル作成されたプログラムは、実動環境で処理する標準的データを操作する際に制御フロー・パスが実行される頻度に応じて、再編成されます。

### 3.1.2 必要な PTF

ANZOBJCVN CL コマンドは、V5R3 用または V5R4 用の PTF として配布されます。この機能を使用するためには、MRI PTF を発注する必要があります。使用する言語用の MRI PTF を発注すると、必要なコード PTF も発注されます。例えば、英語 (2924) の場合、V5R3 用の MRI PTF SI28425、または V5R4 用の MRI PTF SI28415 を発注します。V5R3 および V5R4 用の MRI PTF の全リストは、59 ページの付録 A、『MRI PTF』を参照してください。

**重要：**2008 年 4 月 10 日より前に MRI PTF を発注した場合は、最新のコード PTF (V5R3 用は SI30809、V5R4 用は SI30802) を持っているかどうかを確認してください。最新でない場合は、最新のコード PTF を別途発注してください。

最新のコード PTF を使用するために、以前に行った収集を再実行する必要はありません。ただし、1 プロセッサ以下より小さい区画を分析する場合は、収集を再実行すると、より良い結果が得られます。

また、2007 年 12 月より前に MRI PTF を発注した場合は、少なくとも 2 番目に新しいコード PTF (V5R3 用は SI29369、V5R4 用は SI29370) を使用してデータを収集したかどうかを確認してください。そうでなかった場合は、前述のように、最新のコード PTF を別途発注してください。この場合、以下のいずれかの状態に該当する場合は、収集を再実行すると、より良い結果が得られます。

- ▶ ご使用の 5xx モデル・システムが V6R1 でサポートされると認識されなかった場合。
- ▶ 変換されないと識別された、SQL プロシージャまたは LANGUAGE=JAVA のユーザー定義関数がある場合。
- ▶ パック・データ型を使用して定義されたフィールド値に基づいて QUERY を実行する予定の場合。

表 3-1 は、V6R1 上で作成されたオブジェクトをそれ以前の i5/OS リリースに復元するために必要な PTF をリストしています。

表 3-1 V6R1 上で作成されたオブジェクトをそれ以前の i5/OS リリースに復元するために必要な PTF

i5/OS リリース	必要な PTF
V5R3M0	MF41354
V5R3M5	MF41734
V5R4M0	MF40520
V5R4M5	MF42655

表 3-2 は、V6R1 上で作成された Java プログラムをそれ以前の i5/OS リリースに復元するために必要な PTF をリストしています。

表 3-2 V6R1 上で作成された Java プログラムをそれ以前の i5/OS リリースに復元するために必要な PTF

i5/OS リリース	必要な PTF
V5R3M0	SI27297
V5R4M0	SI27765

### 3.1.3 コマンドの使用上の注意

以下は、ANZOBJCVN CL コマンドの使用に関する注意事項です。

- ▶ このコマンドを実行するには、全オブジェクト (\*ALLOBJ) 特殊権限が必要です。
- ▶ このコマンドは、特に \*COLLECT ステージで、長時間かかることがあります。そのため、このコマンドはバッチ・ジョブで実行することをお勧めします。
- ▶ このコマンドによって参照される 1 次または 2 次独立 ASP (ASP 33-255) をすべてオンに構成変更してください。このコマンドを実行する前に、状況が「Available」であることを確認してください。
- ▶ このコマンドを実行する前に、このコマンドによって参照されるすべてのユーザー定義ファイル・システムをマウントしてください。
- ▶ このコマンドは、次の 2 つの方法のいずれかで実行することを意図しています。
  - 1 つだけのアクティブ・ジョブで ANZOBJCVN を実行し、ライブラリーとディレクトリーを同時に処理する。
  - 同時に 2 つのアクティブ・ジョブで ANZOBJCVN を実行し、1 つのジョブはライブラリーに対してのみ ANZOBJCVN を実行し、もう 1 つはディレクトリーに対してのみ ANZOBJCVN を実行する。
- ▶ 収集されたデータを含むファイルは QUSRSYS ライブラリーにあるので、ANZOBJCVN を使用して情報を収集した後で QUSRSYS ライブラリーを削除すると、情報は失われます。ANZOBJCVN コマンドによって使用されるファイルについての詳細は、23 ページの 3.2.3、『データ収集に使用されるファイル』を参照してください。

**重要：** ANZOBJCVN コマンドを実行する場合、ライブラリー・リスト内で QUSRSYS ライブラリーが QSYS ライブラリーより前にこないようにしてください。そうしないと、ANZOBJCVN コマンドは失敗します。

### 3.1.4 変換時間の見積もり

さまざまな要因によって、プログラム・モデル・オブジェクトの変換にかかる時間の見積もりは、あまり正確な近似値が得られない場合があります。2つの重要な要因は、変換を実行する区画で使用できるハードウェア機能と、ユーザーが選択する変換方式です。

デフォルトでは、変換時間の見積もりは、大体、ANZOBJCVN コマンドが実行される区画の処理容量に基づきます。最初の呼び出し時に変換する方式では、一度に1つのシングル・プロセッサだけを使用する場合があります。したがって、少なくとも1つのプロセッサ全体が割り当てられている区画の場合、見積もりは1つのシングル・プロセッサのCPWレーティングに基づいて行われます。このことは、複数のプロセッサを持つ区画上のV6R1で、異なる変換方式（例えば、STROBJCVN コマンド）を使用した場合、変換時間は見積もりより短くなる可能性があることを意味しています。1つのプロセッサ全体より少ない割り当ての区画の場合、見積もりは、その区画で現在使用可能なプロセッサの部分が考慮されます。

#### 異なるシステムの場合の変換時間の見積もり

デフォルトでは、ANZOBJCVN コマンドの変換時間の見積もりは、そのコマンドが実行されているシステム・モデルに基づきます。異なるシステム・モデルの変換時間を見積もるには、QUSRSYS ライブラリー内にQIZAFTCD と呼ばれるデータ域を作成します。このデータ域に、受動システム・モデルのプロセッサ・フィーチャー・コードを含める必要があります。また、使用されるプロセッサの数も、.01 単位で指定できます。ただし、時間の見積もりに影響を与えるのは、1 プロセッサ以下より少ない割合が指定されている場合だけです。

データ域を作成するには、データ域作成 (CRTDTAARA) CL コマンドを使用します。例：  
CRTDTAARA DTAARA(QUSRSYS/QIZAFTCD) TYPE(\*CHAR) LEN(10) VALUE('7456 0050')

この例では、1つのプロセッサの半分 (.5) が割り当てられた、プロセッサ・フィーチャー・コード 7456 が、変換時間の見積もりのベースとして使用されます。

システムのプロセッサ・フィーチャー・コードを表示するには、以下の手順で行います。

1. 次のハードウェア資源表示 (DSPHDWRSC) CL コマンドを入力して、実行キーを押します。

```
DSPHDWRSC TYPE(*PRC)
```

2. 「プロセッサ資源表示」画面 ( 図 3-1) で、「主カード格納装置」資源の横にオプション 7 ( 資源明細の表示 ) を入力し、実行キーを押します。

プロセッサ資源の表示				システム : RCHAS10
オプションを入力して、実行キーを押してください。				
7= 資源明細の表示				
Opt	資源	タイプ型式	状況	テキスト
7	CEC01	9406-520	操作可能	主カード格納装置
	PN01	28E5	操作可能	システム・コントロール・パネル
	MP01	5229	操作可能	システム・プロセッサ・カード
	MP02	5229	操作可能	システム・プロセッサ・カード
	PV01	7456	操作可能	プロセッサ容量カード
	SP01	28D7	操作可能	サービス・プロセッサ・カード
	BCC01		操作可能	バス・アダプター
	BCC02	5229	操作可能	バス・アダプター
	BCC03	28E7	操作可能	バス・アダプター
	BCC04	28E7	操作可能	バス・アダプター
	BCC05	5229	操作可能	バス・アダプター
	BCC06	5229	操作可能	バス・アダプター
	MS01	30D5	操作可能	2048MB 主記憶域カード
	MS02	30D5	操作可能	2048MB 主記憶域カード
	MS03	30D5	操作可能	2048MB 主記憶域カード
				続く ...
F3= 終了    F5= 最新表示    F6= 印刷    F12= 取り消し				

図 3-1 プロセッサ資源の表示

3. 「資源明細の表示」画面 ( 図 3-2) に、プロセッサ・フィーチャーが表示されます。この例では、プロセッサ・フィーチャー・コードは 7456 です。

資源明細の表示		システム : RCHAS10
資源名	CEC01	
テキスト	主カード格納装置	
タイプ型式	9406-520	
製造番号	10-8A36C	
部品番号		
<b>プロセッサ機構コード</b>	7456	
プロセッサ機構	8955	
位置 :	U9406.520.108A36C	
		終わり
続行するには、実行キーを押してください。		
F3= 終了    F5= 最新表示    F6= 印刷    F12= 取り消し		

図 3-2 資源明細の表示



QIZAFTCD データ域が作成されていないか、不正なデータを含んでいる場合、ANZOBJCVN コマンドは、デフォルトで、そのコマンドが実行されているシステム・プロセッサに基づいて、変換時間を見積もります。

ANZOBJCVN コマンドが実行されているシステムが V6R1 にアップグレードできず、しかもデータ域が存在しないか、不正なデータを含んでいる場合には、表 3-3 に表示されているプロセッサ・タイプのデフォルト値を使用して、変換時間を見積もります。このデフォルト値は、1 プロセッサ全体を想定しています。

表 3-3 QIZAFTCD データ域のデフォルト・フィーチャー・コード

タイプ	型式	フィーチャー・コード
520	0904	7454

## 3.2 情報の収集

ANZOBJCVN コマンドには、情報の収集と報告という 2 つの機能があります。最初にコマンドを使用して情報を収集し、次に報告パラメーターを使用して、前に収集した情報をリストしたスプール・ファイルを生成することができます。このセクションでは、システムからデータを収集する方法について説明します。

**変換時間** : ANZOBJCVN コマンドで報告される変換時間を見積もりです。実際の変換時間は見積もりとは異なる場合があります。CPU の速度、ジョブの優先順位、接続されたディスク・アームの数、使用可能なメモリー、プロセッサの数など、さまざまな要因が変換時間に影響を与える可能性があります。

収集プロセスを開始するには、以下の手順で行います。

1. i5/OS コマンド行から、ANZOBJCVN コマンドを入力し、F4 キーを押してコマンドのプロンプト画面を表示します。
2. オブジェクト変換分析 (ANZOBJCVN) 画面 (図 3-3) で、「オプション」フィールドに \*collect と入力して、実行キーを押します。

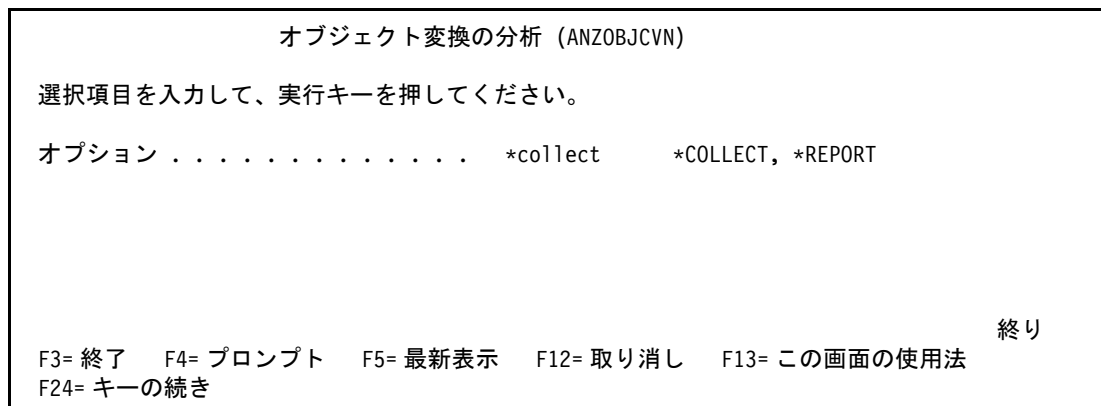


図 3-3 オブジェクト変換分析 (ANZOBJCVN) CL コマンドのプロンプト画面

図 3-4 のように、情報収集のオプションが提供されます。

オブジェクト変換の分析 (ANZOBJCVN)		
選択項目を入力して、実行キーを押してください。		
オプション . . . . .	> *COLLECT	*COLLECT, *REPORT
分析するライブラリー . . . . .	*ALLUSR	名前、総称 *, *ALLUSR...
ASP 装置 . . . . .	*	名前、*, *ALLAVL...
スプールされたファイルの検査 . . . . .	*YES	*YES, *NO
オブジェクト . . . . .	*NONE	
ディレクトリー・サブツリー . . . . .	*ALL	*ALL, *NONE
ターゲット・リリース . . . . .	V6R1M0	VxRyMz
終り		
F3= 終了    F4= プロンプト    F5= 最新表示    F12= 取り消し    F13= この画面の使用法		
F24= キーの続き		

図 3-4 ANZOBJCVN\*COLLECT パラメーター

### 3.2.1 ANZOBJCVN \*COLLECT コマンドのパラメーター

このセクションでは、18 ページの図 3-4 に表示された情報収集のためのさまざまなパラメーターについて説明します。

#### ▶ 分析するライブラリー (LIB)

システム上のライブラリーに対して情報収集が実行されます。ライブラリー内のプログラム・モデル・オブジェクトが報告の対象です。特定のライブラリーから情報を収集するか (この場合は、ライブラリー名を指定します)、または通常のワイルドカード規則を使用してライブラリーの総称リストから情報を収集することを選択できます。また、システムの \*ALLUSR エlementに含まれるすべてのライブラリーを対象にしてデータ収集することも可能です。

\*ALLUSR パラメーターを使用すると、すべてのユーザー・ライブラリーが分析されます。文字 Q で始まっていない名前を持つすべてのライブラリーが分析されます。ただし、以下のライブラリーは除きます。

- #CGULIB
- #COBLIB
- #DFULIB
- #DSULIB
- #RPGLIB
- #RPGLIB
- #SDALIB
- #SEULIB

以下の Qxxx ライブラリーは IBM 提供のものですが、通常は頻繁に変更されるユーザー・データを含んでいます。そのため、これらのライブラリーはユーザー・ライブラリーと見なされ、これらも分析されます。

- QGPL
- QGPL38
- QMGTC
- QMGTC2
- QMPGDATA
- QMQMATA

- QMQMPROC
- QPFRDATA
- QRCL
- QSRVAGT
- QS36F
- QUSER38
- QUSRADSM
- QUSRBRM
- QUSRDIRCL
- QUSRDIRDB
- QUSRINFSKR
- QUSRNOTES
- QUSROND
- QUSRPOSGS
- QUSRPOSSA
- QUSRPYMSVR
- QUSRRDARS
- QUSRSYS
- QUSRVI

**QSYS2\* と SYSIB\*:** ライブラリー QSYS2\* と SYSIB\* は通常、ライブラリーの \*ALLUSR リストに含まれています。したがって、その内容は \*ALLUSR 保存操作に含まれます。しかし、i5/OS のアップグレード時にこれらのライブラリー内のプログラムが置き換えられるため、ANZOBJCVN コマンドは、\*ALLUSR が指定された場合、これらのライブラリーの内容を検査しません。

**IBM プロダクト・ライブラリー:** 個々の IBM プロダクト・ライブラリーに関する変換情報を収集することもできます。IBM スキップ・シップ・プロダクト (V6R1 用の新規バージョンが提供されないプロダクト) のライブラリーは、71 ページの付録 C、『変換を必要とする IBM ソフトウェア・スキップ・シップ・プロダクト』にリストされています。

このパラメーターには \*NONE を指定することもできます。この場合は、どのライブラリーも分析されません。統合ファイル・システム・ディレクトリーにあるオブジェクトだけを分析したい場合は、LIB(\*NONE) を指定できます。

▶ **ASP 装置 (ASPDEV)**

このパラメーターは、分析対象のライブラリーやスプール・ファイルのストレージが割り振られている ASP 装置を指定します。ライブラリーを格納する ASP が、スレッドのライブラリー・ネーム・スペースの一部に含まれていない場合は、必ずこのパラメーターを指定して、正しいライブラリーが分析されるようにする必要があります。

**オブジェクト・パラメーターの場合は無視:** このパラメーターは、オブジェクト (OBJ) パラメーター (後述) で指定されたオブジェクトに対しては適用されません。オブジェクトのパス名の一部として、独立 ASP (IASP) 名を含んでいるからです。

- **\*(アスタリスク)**

ライブラリーを見つけるために、現在スレッドのライブラリー・ネーム・スペースの一部として含まれている ASP が検索されます。これには、システム ASP (ASP 1)、すべての定義済み基本ユーザー ASP (ASP 2-32)、および、スレッドに ASP グループがある場合は、スレッドの ASP グループ内の 1 次および 2 次 ASP が含まれます。

- **\*ALLAVL**

すべての使用可能な ASP が検索されます。これには、システム ASP (ASP1)、すべての定義済み基本ユーザー ASP (ASP 2-32)、およびすべての使用可能な 1 次および 2 次 ASP (ASP 33-255 のうち状況が「Available」のもの) が含まれます。

– \*CURASPGRP

スレッドに ASP グループがある場合、スレッドの ASP グループ内の 1 次および 2 次 ASP を検索して、ライブラリーを見つけます。システム ASP (ASP 1) と定義済み基本ユーザー ASP (ASP 2-32) は検索されません。スレッドに ASP グループが何も関連付けられていない場合、エラー・メッセージが発行されます。

– \*SYSBAS

ライブラリーを見つけるために、システム ASP (ASP 1) とすべての定義済み基本ユーザー ASP (ASP 2-32) が検索されます。スレッドに ASP グループがある場合でも、1 次または 2 次 ASP は検索されません。

– 名前

「名前」オプションを使用して、検索対象の 1 次または 2 次 ASP 装置の名前を指定することができます。1 次または 2 次 ASP は、活動状態であり (ASP 装置をオンに構成変更することによって活動化します)、状況が「Available」でなければなりません。システム ASP (ASP 1) と定義済み基本ユーザー ASP (ASP 2-32) は検索されません。

▶ スプール・ファイルの検査 (SPLFILE)

このパラメーターは、各 ASP 上のスプール・ファイルを変換するために必要な時間を見積もるかどうかを指定します。SPLFILE(\*YES) を指定した場合、LIB(\*NONE) は指定できません。次の 2 つのオプションが可能です。

- \*YES: 各 ASP 上のスプール・ファイルが分析されます。
- \*NO: 各 ASP 上のスプール・ファイルは分析されません。

**注:** 選択された ASP 装置 (ASPDEV) は、スプール・ファイル・データの収集に影響を与えます。

スプール・ファイルについて詳しくは、43 ページの 3.7.3、『スプール・ファイル』を参照してください。

▶ オブジェクト (OBJ)

このパラメーターは、分析される統合ファイル・システム・オブジェクトのパス名、または名前に突き合わせるパターンを指定します。オブジェクトのパス名は、単純名でも、オブジェクトが置かれているディレクトリー名で修飾された名前でも構いません。パターンは、パス名の最後の部分で指定できます。アスタリスク (\*) は任意の文字数に突き合わせ、疑問符 (?) は単一文字に突き合わせます。パス名が修飾されているか、パターンを含んでいる場合、パス名をアポストロフィで囲む必要があります。

**パターン・マッチング:** オブジェクト (OBJ) パラメーターのパターン・マッチングは、第 1 レベルのオブジェクトにのみ適用されます。第 1 レベルのオブジェクトがディレクトリーである場合、パターン・マッチングは、ディレクトリーの内容またはそのサブディレクトリーの内容には適用されません。

**シンボリック・リンク:** パス名の最後の構成要素がシンボリック・リンクである場合、そのシンボリック・リンク・オブジェクトは分析されます。しかし、そのシンボリック・リンクによって指し示されているオブジェクトは分析されません。

\*NONE を指定すると、どの統合ファイル・システム・オブジェクトも分析されないことを意味します。

▶ ディレクトリー・サブツリー (SUBTREE)

このパラメーターは、オブジェクト (OBJ) パラメーターで指定されたオブジェクトがディレクトリーである場合、そのサブツリー内のオブジェクトを分析するかどうかを指定します。パラメーター \*ALL が使用された場合、オブジェクト (OBJ) パラメーターで指定されたオブジェクトが、必要に応じて分析されます。オブジェクトがディレクトリーである場合、その内容だけでなく、そのサブディレクトリーのすべての内容も分析されます。

**注:** このパラメーターは、統合ファイル・システム内のオブジェクトにのみ適用されます。OBJ パラメーターが指定されている場合にのみ有効です。

コマンドが特定のディレクトリーのサブツリーの処理を開始すると、その後に検出されて処理されるオブジェクトは、指定のディレクトリー・ツリー内のオブジェクトの編成を更新する操作によって影響を受けることがあります。これには、以下の操作が含まれます。

- オブジェクト・リンクの追加、削除、または名前変更
- ファイル・システムのマウントまたはアンマウント
- コマンドを呼び出すプロセスの有効ルート・ディレクトリーの更新
- シンボリック・リンクの内容の更新

ディレクトリーのサブツリーを処理するために、システム・コードは、処理中にジョブ内で開くことができるファイル記述子の最大数を増やすことがあります。これは、記述子の不足が原因でコマンドが失敗することがないようにするために行われます。このジョブ有効範囲内の最大値は、コマンドが完了してもリセットされません。

\*NONE を指定することができ、この場合は、オブジェクト (OBJ) パラメーターで指定されたオブジェクトのみが分析されます。オブジェクトがディレクトリーである場合、ディレクトリーは分析されますが、その内容は分析されません。

▶ ターゲット・リリース (TGTRLS)

このパラメーターは、データまたはプログラムのマイグレーション先またはアップグレード先のリリースを指定します。この情報を使用して、実行する具体的な分析が決まります。V6R1 が、現在サポートされる唯一のターゲット・リリースです。

### 3.2.2 収集コマンドの例

データを収集するための 2 つのアプローチのうち、1 つの方法について説明します。1 つの方法は、図 3-5 に示すような ANZOBJCVN コマンドを使用して、すべてのタイプの情報を一度に収集し、すべての収集ファイルにデータを追加することです。このコマンドは、システムの \*ALLUSR 部分のライブラリーで検出されるすべてのプログラム・オブジェクト (18 ページの 3.2.1、『ANZOBJCVN \*COLLECT コマンドのパラメーター』) にリストされている例外を除く)、スプール・ファイル、および統合ファイル・システム内のすべての関連オブジェクトについての情報を収集します。

オブジェクト変換の分析 (ANZOBJCVN)		
選択項目を入力して、実行キーを押してください。		
オプション . . . . .	> *COLLECT	*COLLECT, *REPORT
分析するライブラリー . . . . .	*ALLUSR	名前、総称 *, *ALLUSR...
ASP 装置 . . . . .	*	名前、*, *ALLAVL...
スプールされたファイルの検査 . . . . .	*YES	*YES, *NO
オブジェクト . . . . .	> /	
ディレクトリーのサブツリー . . . . .	*ALL	*ALL, *NONE
ターゲット・リリース . . . . .	V6R1M0	VxRyMz
終り		
F3= 終了   F4= プロンプト   F5= 最新表示   F12= 取り消し   F13= この画面の使用法		
F24= キーの続き		

図 3-5 ANZOBJCVN による \*ALLUSR ライブラリー、統合ファイル・システム、およびスプール・ファイルのデータ収集

代わりに、データ収集プロセスを 2 つのステップに分けることもできます。例えば、\*ALLUSR ライブラリー内のプログラム・モデル・オブジェクトとスプール・ファイルに関するデータを収集できます。

```
ANZOBJCVN OPTION(*COLLECT) LIB(*ALLUSR) SPLFILE(*YES)
```

この収集の後に、ディレクトリー・オブジェクトの収集を行います。

```
ANZOBJCVN OPTION(*COLLECT) LIB(*NONE) SPLFILE(*NO) OBJ(Ð/Ð)
```

**重要：**上記の ANZOBJCVN コマンドを統合ファイル・システム全体に対して実行した場合、診断メッセージ CPDB0C8 が /dev/QASPO1 フォルダーに出されることが予想されません。このメッセージは、OBJ('/') を指定した場合に生成され、このパスはスキップされます。

データを収集した後、ANZOBJCVN の報告オプションのいずれかを使用して、結果を表示することができます。23 ページの 3.3、『報告書の作成』を参照してください。

個別のライブラリー、統合ファイル・システムのサブセット、あるいはその両方を同時に対象にして、データを収集することもできます。ただし、この方法を選択した場合は、必ず報告書を作成して検討してから、次のライブラリーまたはファイル・システムの別のサブセットのデータ収集を続行してください。特定のタイプの収集を実行するたびに、そのタイプの情報を保管するために使用されるファイル内のデータが置き換えられます。代わりに、ライブラリー内のプログラムと同様に、あるタイプの収集を実行するたびに、収集ファイルを別のファイル・セットにコピーすることもできます。その後で、各コピー・ファイルに対して QUERY を出すことができます。23 ページの 3.2.3、『データ収集に使用されるファイル』、および 35 ページの 3.5、『分析データに対する QUERY』を参照してください。

次の例は、1 つの特定のライブラリー内のプログラムに関する情報を収集する方法を示しています。

```
ANZOBJCVN OPTION(*COLLECT) LIB(myLib) SPLFILE(*NO)
```

次の例は、統合ファイル・システム内のディレクトリーのサブセットからの収集を示しています。

```
ANZOBJCVN OPTION(*COLLECT) LIB(*NONE) SPLFILE(*NO) OBJ(Ð/myPartÐ) SUBTREE(*ALL)
```

ライブラリーの収集とディレクトリー・オブジェクトの収集を組み合わせることも可能です。この場合は、1つのライブラリーと1つのディレクトリーのみを対象とします。

```
ANZOBJCVN OPTION(*COLLECT) LIB(aLib) SPLFILE(*NO) OBJ(Ⓓ/ⒶPathⒹ) SUBTREE(*NONE)
```

個々の IBM プロダクト・ライブラリーに関する変換情報を収集することもできます。IBM スキップ・シップ・プロダクト (V6R1 用の新規バージョンが提供されないプロダクト) のライブラリーは、71 ページの付録 C、『変換を必要とする IBM ソフトウェア・スキップ・シップ・プロダクト』にリストされています。

### 3.2.3 データ収集に使用されるファイル

ANZOBJCVN コマンドによって収集されるデータは、QUSRSYS ライブラリー内の 4 つの異なるファイルに保管されます。

- ▶ QSYS 内の QAIZACVN ファイルは、マスター・テンプレート・ファイルで、収集されたデータは追加されません。ただし、QUSRSYS 内の QAIZACVN は、ライブラリー内の分析データの収集のターゲット・ファイルです。
- ▶ QAIZADIR ファイルは、オブジェクト・パラメーターが使用された場合のディレクトリー情報のターゲットです。
- ▶ QAIZAOBJ ファイルは、オブジェクト・パラメーターが使用された場合の統合ファイル・システム・オブジェクト情報のターゲットです。
- ▶ QAIZASPL ファイルは、スプール・ファイルの収集情報のターゲットです。

それぞれのファイルには、そのタイプの最後の収集からのデータのみが入っています。後続の収集は、現在これらのファイルに保持されているデータを置き換えることとなります。特定のタイプについて一度にすべてのデータを収集するのでない限り (プログラム・モデル・オブジェクトのように)、後で収集を分析するために、これらのファイルを別のライブラリーにコピーしておくことも可能です。必要に応じて、別のライブラリーに移動された収集を対象にして QUERY を実行し、追加の報告書を生成することができます。QUERY の作成についての詳細は、35 ページの 3.5、『分析データに対する QUERY』を参照してください。ANZOBJCVN OPTION(\*REPORT) コマンドは、QUSRSYS ライブラリー内のファイルに対してのみ実行されます。他のライブラリー内にあるコピーされた収集ファイルに対しては実行されません。

**重要:** ANZOBJCVN コマンドを実行する場合、ライブラリー・リスト内で QUSRSYS ライブラリーが QSYS ライブラリーより前にこないようにしてください。そうしないと、ANZOBJCVN コマンドは失敗します。

## 3.3 報告書の作成

ANZOBJCVN CL コマンドには、情報の収集と報告という 2 つの機能があります。最初にコマンドを使用して情報を収集します。次に、報告パラメーターを使用して、前に収集した情報をリストしたスプール・ファイルを生成することができます。このセクションでは、ANZOBJCVN OPTION(\*REPORT) コマンドを使用して報告書を生成する方法について説明します。OPTION(\*REPORT) を指定して ANZOBJCVN コマンドを実行すると、このツールは ANZOBJCVN OPTION(\*COLLECT) コマンドを使用して最後に収集されたデータのタイプに基づいて、QPIZARPT と呼ばれるスプール・ファイルに報告書を生成します。

QPIZARPT ファイルは、出荷時にはレコードの最大数が制限されています。ユーザーがすべての使用可能な報告書 (詳細報告書を含む) を印刷することを選択した場合、あるいは 1 つのオブジェクト要約報告書を印刷する場合でも、それが大きいファイル・システムを含ん

でいる場合、報告書はレコードの最大数を超える可能性があります。CPA4072 オペレーター照会メッセージに応答する必要が生じるのを避けるために、報告書を印刷する前に、次の印刷装置ファイル変更 (CHGPRTF) コマンドを使用して、レコードの最大数を変更することができます。

CHGPRTF FILE(QSYS/QPIZARPT) MAXRCDS(\*NOMAX)

**変換時間** : ANZOBJCVN コマンドで報告される変換時間は見積もりです。実際の変換時間は見積もりとは異なる場合があります。CPU の速度、ジョブの優先順位、接続されたディスク・アームの数、使用可能なメモリー、プロセッサの数など、さまざまな要因が変換時間に影響を与える可能性があります。

報告プロセスを開始するには、以下の手順で行います。

1. i5/OS コマンド行から、ANZOBJCVN CL コマンドを入力し、F4 キーを押してコマンドのプロンプト画面を表示します。
2. オブジェクト変換分析 (ANZOBJCVN) 画面 (図 3-6) で、「オプション」フィールドに \*report と入力して、実行キーを押します。

オブジェクト変換の分析 (ANZOBJCVN)

選択項目を入力して、実行キーを押してください。

オプション . . . . . \*report            \*COLLECT, \*REPORT

終り

F3= 終了    F4= プロンプト    F5= 最新表示    F12= 取り消し    F13= この画面の使用法  
F24= キーの続き

図 3-6 オブジェクト変換分析 (ANZOBJCVN) CL コマンドのプロンプト画面

図 3-7 のように、情報を報告するためのオプションが提供されます。

オブジェクト変換の分析 (ANZOBJCVN)

選択項目を入力して、実行キーを押してください。

オプション . . . . . > \*REPORT            \*COLLECT, \*REPORT  
報告書のタイプ . . . . . \*ALLAVL            \*ALLAVL, \*CVNPRB, \*LIBSUM...

値の続きは + \_\_\_\_\_

終り

F3= 終了    F4= プロンプト    F5= 最新表示    F12= 取り消し    F13= この画面の使用法  
F24= キーの続き

図 3-7 ANZOBJCVN \*REPORT コマンド

**注** : ANZOBJCVN OPTION(\*REPORT) を指定した場合、他のパラメーターは使用されません。F9 を使用して前の ANZOBJCVN OPTION(\*COLLECT) コマンドを再呼び出しすると、ユーザーは追加パラメーターが使用される可能性があることに気付くはずですが、しかし、実際には、図 3-7 に示されているパラメーターのみが使用されます。



### 3.3.1 ANZOBJCVN \*REPORT コマンドのパラメーター

図 3-7 に表示されている「報告書のタイプ」パラメーターには、報告書を生成するための以下のオプションが含まれます。

▶ \*ALLAVL

各タイプの最新の収集で指定されたライブラリーおよびディレクトリー内のすべてのオブジェクトについて、使用可能なすべての要約報告書および詳細報告書が生成されます。直前のライブラリーの収集で、スプール・ファイルの検査 (SPLFILE) パラメーターが \*YES に設定されていた場合、スプールされたデータベース・メンバーの情報が、ライブラリー要約報告書内に生成されます。

▶ \*CVNPRB

最新のライブラリー収集で指定された各ライブラリーについて、変換されないプログラム・オブジェクトの詳細報告書が生成されます。

▶ \*LIBSUM

最新のライブラリー収集で指定された各ライブラリーについて、要約報告書が生成されます。直前の収集時に、スプール・ファイルの検査 (SPLFILE) パラメーターが \*YES に設定されていた場合、スプールされたデータベース・メンバーの情報もこの報告書内に生成されます。

▶ \*LIBDTL

最新のライブラリー収集で指定された各ライブラリーについて、詳細報告書が生成されます。

▶ \*OBJSUM

最新のオブジェクト収集で指定された各統合ファイル・システム・ディレクトリーについて、要約報告書が生成されます。

▶ \*OBJDTL

最新のオブジェクト収集で指定された各統合ファイル・システム・オブジェクトについて、詳細報告書が生成されます。

ヒント : ANZOBJCVN OPTION(\*REPORT) コマンドを実行する際には、以下のヒントを使用してください。

- ▶ 報告書を要求する前に、該当の収集を完了している必要があります。そうしないと、メッセージ CPF80DC が生成され、理由コード 3 が示されます。
- ▶ 収集が完了したが、報告するデータがない場合、例えば、\*CVNPRB 報告書が要求されたときに、変換に失敗するプログラムが何もない場合、メッセージ CPI1E10「表示する印刷装置出力はない (No printer output to display)」が生成されます。
- ▶ 報告書の生成時に「報告書のタイプ」パラメーターで正符号 (+) を入力して、複数のパラメーターを結合することができます。

最初に、\*CVNPRB パラメーターと \*LIBSUM パラメーターを使用することをお勧めします。この 2 つの報告書は、注意が必要なプログラム・モデル・オブジェクトと、プログラムおよびスプール・ファイルの変換時間の要約見積もりのみを表示します。一般に、この 2 つの報告書で、V6R1 へのアップグレードを準備するために必要なすべての情報が提供されるはずですが。

これに対して、詳細報告書は各オブジェクトに関する情報を表示します。そのため、\*LIBDTL または \*OBJDTL パラメーターを使用して生成される報告書は、非常に大きいスプール・ファイル出力を生成することがあります。\*OBJSUM 報告書も、データ収集の対象となった各サブディレクトリーに関する情報を表示するので、非常に大きくなる場合があります。

まず、詳細報告書は、特定のプログラムに必要な変換時間や、影響を受ける可能性があるファイル・システムのファイル名などの情報を提供しますが、詳細報告書を印刷する代わりに、QUERY を実行することもできます。35 ページの 3.5、『分析データに対する QUERY』を参照してください。

### 3.3.2 報告書コマンドの例

以下の例は、ANZOBJCVN OPTION(\*REPORT) コマンドの使用法を示しています。

ANZOBJCVN OPTION(\*REPORT) RPTTYPE(\*CVNPRB \*LIBSUM)

この例では、プログラム変換の問題 (\*CVNPRB) 報告書とライブラリー要約 (\*LIBSUM) 報告書が生成されます。図 3-8 は、このセクションの初めに示した ANZOBJCVN OPTION(\*REPORT) コマンドから生成されたプログラム変換の問題 (\*CVNPRB) 報告書の最初の 2 ページを示しています。

オブジェクト変換の分析 - プログラム変換の問題 10/31/07 15:49:57							ページ	1
5722SS1 V5R4M0 060210 RCHAS61								
収集の選択基準 - ANZOBJCVN OPTION(*COLLECT)								
ライブラリー		*ALLUSR						
ASPDEV		*						
SPLFILE		*YES						
オブジェクト		/						
サブツリー		*ALL						
ターゲット・リリース		V6R1M0						
報告書の選択基準 - ANZOBJCVN OPTION(*REPORT)								
報告書タイプ		*CVNPRB						
ライブラリー		*ALLUSR						
ASPDEV		*						
SPLFILE		*YES						
オブジェクト		/						
サブツリー		*ALL						
ターゲット・リリース		V6R1M0						
オブジェクト変換の分析 - プログラム変換の問題 10/31/07 15:49:57							ページ	2
5722SS1 V5R4M0 060210 RCHAS61								
プロセッサ・フィーチャー・コード . . . . . : 7457								
補助記憶域プール (ASP) . . . . . : 00001 ASP 資源名 : *SYSBAS								
----- カウンター -----								
		合計		変換				
ライブラリー		オブジェクト		不能				
-----								
ORBCOLL		465		1				
オブジェクト	タイプ	システム	作成	デジタル	プロファイル	状態		
		レベル	データ	署名入り				
-----								
BRANCHCNTR	PGM	V4R5M0	NO	NO	NO	*USER		
補助記憶域プール (ASP) . . . . . : 00001 ASP 資源名 : *SYSBAS								

図 3-8 ANZOBJCVN プログラム変換の問題報告書

図 3-9 は、オブジェクト要約報告書 (\*OBJSUM) の最初の 2 ページを示しています。

オブジェクト変換の分析 - オブジェクト要約報告書		10/31/07 15:52:45
5722SS1 V5R4M0 060210 RCHAS61		ページ 1
収集の選択基準 - ANZOBJCVN OPTION(*COLLECT)		
ライブラリー	*ALLUSR	
ASPDEV	*	
SPLFILE	*YES	
オブジェクト	/	
サブツリー	*ALL	
ターゲット・リリース	V6R1M0	
報告書の選択基準 - ANZOBJCVN OPTION(*REPORT)		
報告書タイプ	*OBJSUM	
ライブラリー	*ALLUSR	
ASPDEV	*	
SPLFILE	*YES	
オブジェクト	/	
サブツリー	*ALL	
ターゲット・リリース	V6R1M0	
プロセッサ・フィーチャー・コード . . . . .		7457
収集された日付 . . . . .		10/31/07
変換が必要なオブジェクト数 . . . . .		156
ディレクトリー・オブジェクトの推定変換時間合計 . . . . .		00:07:15
オブジェクト変換の分析 - オブジェクト要約報告書		10/31/07 15:52:45
5722SS1 V5R4M0 060210 RCHAS61		ページ 2
ディレクトリー : /		
--- カウンター ---	----- 症状 -----	
変換 必要	Unicode 文字	変換 時間
-----	-----	-----
0	0	00:00:00
ディレクトリー : /QopenSys		
--- カウンター ---	----- 症状 -----	
変換 必要	Unicode 文字	変換 時間
-----	-----	-----
0	0	00:00:00
ディレクトリー : /QopenSys/QIBM		
--- カウンター ---	----- 症状 -----	
変換 必要	Unicode 文字	変換 時間
-----	-----	-----
0	0	00:00:00
ディレクトリー : /QopenSys/QIBM/ProdData		
--- カウンター ---	----- 症状 -----	
変換 必要	Unicode 文字	変換 時間
-----	-----	-----
0	0	00:00:00
. . .		

図 3-9 ANZOBJCVN オブジェクト要約報告書

例えば、以下のようなコマンドを入力することにより、ディレクトリー・オブジェクトとライブラリー・オブジェクトの組み合わせに基づいて報告書を生成することも可能です。

ANZOBJCVN OPTION(\*REPORT) RPTTYPE(\*LIBSUM \*OBJSUM)

### 3.3.3 変換の報告書を理解する

このセクションでは、変換報告書について説明し、提供される情報を理解できるようにします。29 ページの図 3-10 に示されている報告書から見ていきます。それぞれの報告書タイプには、2 つのセクションが含まれています。

- ▶ 「*収集の選択基準*」セクションは、報告のために使用可能な、ANZOBJCVN OPTION(\*COLLECT) コマンドで収集された最新データを示します。「ライブラリー」、「ASPDEV」、および「SPLFILE」の各行は、ライブラリーのデータ収集に関連した情報で、ライブラリーのデータ収集が実行されるたびに更新されます。「オブジェクト」、「サブツリー」、および「ターゲット・リリース」の各行は、オブジェクトのデータ収集に関連した情報で、オブジェクトのデータ収集が実行されるたびに更新されます。
- ▶ 「*報告書の選択基準*」セクションは、報告を要求されたデータを示しています。

図 3-10 にある報告書の一部には、以下の情報が表示されています。

- ▶ 報告書が生成された日付
- ▶ i5/OS のバージョンとシステム名
- ▶ 収集された最新のデータ・タイプ (追加情報は、23 ページの 3.2.3、『データ収集に使用されるファイル』を参照)
- ▶ 変換時間の見積もりに使用されたプロセッサ・フィーチャー・コード

報告書の下部にも、以下の事項に関する重要な情報が表示されます。

- ▶ 分析されたオブジェクト数
- ▶ 属性の逸失を伴う、伴わないオブジェクト変換
- ▶ 変換されるオブジェクトの合計数
- ▶ オブジェクトを新規 V6R1 形式に変換するための見積もり時間

この例では、ライブラリー PFREXP で検出された 57 個の適格プログラム・オブジェクトの変換に 00:00:13 時間かかると見積もっています。これは、0 時間、0 分、13 秒を表しています。\*LIBDTL でコンパイルされた報告書に表示される変換時間は、個々のプログラム・オブジェクトをリストし、変換時間を秒数でのみ表示します。生成された報告書では、ある程度の切り上げや切り捨てが行われることが分かります。これは特に、1 つのライブラリー内のすべてのプログラム・オブジェクトの累積の変換時間が 1 秒未満の場合に見られます。この場合、\*LIBSUM 報告書形式のライブラリーの報告書は、多くの場合、00:00:00 として表示されます。

\*LIBSUM 報告書の変換時間は、時間、分、秒の形式 (例えば、HH:MM:SS) で報告されません。個々のプログラム・オブジェクトの \*LIBDTL 報告書の変換時間は、秒数とその小数部分でのみ報告されます。例えば、図 3-10 では、プログラム・オブジェクト BUPAP1CC の変換時間は 0.124 秒です。

オブジェクト変換の分析 - ライブラリー詳細報告書							9/21/07 14:35:05	ページ 1
5722SS1 V5R4M0 060210 RCHAS10								
収集の選択基準 - ANZOBJCVN OPTION(*COLLECT)								
ライブラリー	PFREXP							
ASPDEV	*							
SPLFILE	*YES							
オブジェクト	/							
サブツリー	*ALL							
ターゲット・リリース	V6R1M0							
報告書の選択基準 - ANZOBJCVN OPTION(*REPORT)								
報告書タイプ	*LIBDTL							
ライブラリー	PFREXP							
ASPDEV	*							
SPLFILE	*YES							
オブジェクト	/							
サブツリー	*ALL							
ターゲット・リリース	V6R1M0							
オブジェクト変換の分析 - ライブラリー詳細報告書							9/21/07 14:35:05	ページ 2
5722SS1 V5R4M0 060210 RCHAS10								
プロセッサ・フィーチャー・コード . . . . . : 7456								
補助記憶域プール (ASP) . . . . . : 00001      ASP 資源名 : *SYSBAS								
----- カウンター -----      ----- 症状 -----								
ライブラリー	合計 オブジェクト数	変換 必要	逸失なし 属性	一部逸失 属性	変換 時間			
PFREXP	57	57	57	0	00:00:13			
オブジェクト	タイプ	システム レベル	作成 データ	デジタル 署名入り	プロファイル 状態	変換 見積もり (秒)		
BUPAP1CC	PGM	V3R6M0	YES	NO	NO	*USER 0.124		
BUPAP1CCN	PGM	V3R6M0	YES	NO	NO	*USER 0.124		
BUPAP1CJ	PGM	V3R6M0	YES	NO	NO	*USER 0.105		
BUPAP1CJE	PGM	V3R6M0	YES	NO	NO	*USER 0.115		
BUPAP1IN	PGM	V3R6M0	YES	NO	NO	*USER 0.401		
BUPAP1INDQ	PGM	V3R6M0	YES	NO	NO	*USER 0.716		
BUPAP1INSQ	PGM	V3R6M0	YES	NO	NO	*USER 0.411		
BUPAP1ISEQ	PGM	V3R6M0	YES	NO	NO	*USER 0.391		
BUPAP1ISER	PGM	V3R6M0	YES	NO	NO	*USER 0.391		
BUPAP1I2	PGM	V3R6M0	YES	NO	NO	*USER 0.458		
BUPAP1RCN	PGM	V3R6M0	YES	NO	NO	*USER 0.573		
BUPAP1RN	PGM	V4R4M0	YES	NO	NO	*USER 0.649		
BUPCRTJRN	PGM	V3R6M0	YES	NO	NO	*USER 0.124		

図 3-10 ライブラリー PFREXP の ANZOBJCVN ライブラリー詳細報告書

31 ページの図 3-11 は、この後のライブラリー詳細報告書に表示されている、ライブラリー内の一部のオブジェクトの詳細に関する情報を示しています。各オブジェクトについて、以下の情報が収集され、報告されます。

- ▶ オブジェクト名
- ▶ オブジェクト・タイプ
- ▶ システム・レベル。これは、オブジェクトが保管できる最も古いリリースです。
- ▶ 作成データ。これは、オペレーティング・システムの新規リリースに変換するために必要なすべてのデータをオブジェクトが持っているかどうかを示します。  
N/A の値は「適用外」を意味し、\*SQLPKG オブジェクトと、SQL 用にのみ使用される特定の \*PGM および \*SRVPGM オブジェクトに対して表示されます。これらのオブジェクトは、プログラムとして実行されないため、プログラム変換の必要はありません。
- ▶ デジタル署名入り。これは、オブジェクトがデジタル署名を持つかどうかを示します。  
プログラム変換は、以前にオブジェクトに関連付けられたデジタル署名をすべて除去します。プログラムが変換される前に追加されたデジタル署名は、変換後は無効になるからです。
- ▶ プロファイル。オブジェクトがプロファイル作成の効果を失うかどうかを示します。  
プログラムが以前に適用されたプロファイル作成データを持っていたが、そのカプセル化されたプロファイル作成データが除去された場合、そのプログラムは、プログラム変換時にプロファイル作成の効果を失います。
- ▶ 状態。これは、プログラムの現在の状態を示します。  
現在の状態が \*SYSTEM または \*INHERIT のプログラムは、プログラム変換時に \*USER に変更されます。
- ▶ 見積もり変換時間。プログラム・オブジェクトの変換時間の見積もりを示します。

**ヒント:** デジタル署名、プロファイル作成、およびプログラム状態についての詳細は、12 ページの 3.1.1、『用語の定義』を参照してください。

オブジェクト	タイプ	システム レベル	作成 データ	デジタル 署名入り	プロファイル	状態	変換 見積もり (秒)
BUPAP1CC	PGM	V3R6M0	YES	NO	NO	*USER	0.124
BUPAP1CCN	PGM	V3R6M0	YES	NO	NO	*USER	0.124
BUPAP1CJ	PGM	V3R6M0	YES	NO	NO	*USER	0.105
BUPAP1CJE	PGM	V3R6M0	YES	NO	NO	*USER	0.115
BUPAP1IN	PGM	V3R6M0	YES	NO	NO	*USER	0.401
BUPAP1INDQ	PGM	V3R6M0	YES	NO	NO	*USER	0.716
BUPAP1INSQ	PGM	V3R6M0	YES	NO	NO	*USER	0.411
BUPAP1ISEQ	PGM	V3R6M0	YES	NO	NO	*USER	0.391
BUPAP1ISER	PGM	V3R6M0	YES	NO	NO	*USER	0.391
BUPAP1I2	PGM	V3R6M0	YES	NO	NO	*USER	0.458
BUPAP1RCN	PGM	V3R6M0	YES	NO	NO	*USER	0.573
BUPAP1RN	PGM	V4R4M0	YES	NO	NO	*USER	0.649
BUPCRTJRN	PGM	V3R6M0	YES	NO	NO	*USER	0.124
BUPINZFE	PGM	V4R4M0	YES	NO	NO	*USER	0.220
BUPMUNUNE	PGM	V4R4M0	YES	NO	NO	*USER	0.325
CHAINUP_CW	PGM	V3R6M0	YES	NO	NO	*USER	0.124
CLCSTPEX	PGM	V4R4M0	YES	NO	NO	*USER	0.191
CLCSTPEXB	PGM	V3R6M0	YES	NO	NO	*USER	0.153
CLCSTPEXHI	PGM	V3R6M0	YES	NO	NO	*USER	0.181
CLRPLIBALL	PGM	V4R1M0	YES	NO	NO	*USER	0.258
CLRPLIB20	PGM	V3R6M0	YES	NO	NO	*USER	0.201
CLRPLIB25	PGM	V4R1M0	YES	NO	NO	*USER	0.229

図 3-11 ANZOBJCVN ライブラリー詳細報告書で報告される、ライブラリー PFREXP 内のオブジェクト

### 3.4 変換されないプログラム

変換できないプログラム・オブジェクトがあるかどうかを調べるための推奨される方法は、ANZOBJCVN OPTION(\*REPORT) コマンドの「報告書タイプ」パラメーターに \*CVNPRB 値を使用することです。前の収集で指定された各ライブラリーごとに、変換されないプログラム・オブジェクトの詳細報告書が生成されます。\*CVNPRB パラメーターを使用して生成される報告書の例は、32 ページの図 3-12 を参照してください。この報告書では、ライブラリー MILLSI のプログラム・オブジェクト NOCRTDATA は、必要な作成データが欠落しているために、変換に失敗することを示しています。

QUERY を作成して、収集時に使用されたファイルからデータを抽出することも可能です。変換できないプログラムをリストする QUERY の例は、35 ページの 3.5、『分析データに対する QUERY』を参照してください。

個々のプログラムをチェックするには、プログラム表示 (DSPPGM) またはサービス・プログラム表示 (DSPSRVPGM) CL コマンドを使用して、十分な作成データが存在するかどうかを調べることもできます。ILE (サービス) プログラムの場合、「すべての作成データ (All creation data)」が \*NO 以外の値であれば、十分な作成データが使用可能であることを示します。OPM プログラムの場合、「プログラム識別情報 (Observable information)」が \*NONE 以外の値であれば、十分な作成データが使用可能であることを示します。

さらに、\*LIBDTL パラメーターを使用して作成された詳細なライブラリー変換報告書を検討して、どのプログラム・オブジェクトを変換できないかを判別することもできます。この報告書では、変換できないプログラムの「作成データ」列に「NO」の値が入ります。

**作成データのないプログラム** : ANZOBJCVN コマンドで作成データのないプログラムが報告された場合、そのプログラムがまだ使用されているかどうかを確認してください。例えば、そのプログラムはアプリケーションの以前のバージョンの一部であり、もう使われていない可能性があります。作成データのないプログラムがまだ使われている場合は、再コンパイルが必要です。あるいは、プログラムがソース・コードなしで提供された場合は、作成データを含む新しいバージョンと取り替える必要があります。

オブジェクト変換の分析 - プログラム変換の問題		10/26/07 11:43:50	ページ 1
5722SS1 V5R4M0 060210 RCHAS10			
収集の選択基準 - ANZOBJCVN OPTION(*COLLECT)			
ライブラリー	*ALLUSR		
ASPDEV	*		
SPLFILE	*NO		
オブジェクト	/		
サブツリー	*ALL		
ターゲット・リリース	V6R1M0		
報告書の選択基準 - ANZOBJCVN OPTION(*REPORT)			
報告書タイプ	*CVNPRB		
ライブラリー	*ALLUSR		
ASPDEV	*		
SPLFILE	*NO		
オブジェクト	/		
サブツリー	*ALL		
ターゲット・リリース	V6R1M0		
オブジェクト変換の分析 - プログラム変換の問題		10/26/07 11:43:50	ページ 2
5722SS1 V5R4M0 060210 RCHAS10			
プロセッサ・フィーチャー・コード . . . . .		: 7456	
補助記憶域プール (ASP) . . . . .		: 00001	ASP 資源名 : *SYSBAS
----- カウンター -----			
	合計	変換	
ライブラリー	オブジェクト	不能	
-----	-----	-----	
MILLSI	2	1	
オブジェクト	タイプ	システム レベル	作成 データ
-----	-----	-----	-----
NOCRTDATA	PGM	V4R5M0	NO
補助記憶域プール (ASP) . . . . .			: 00001
			ASP 資源名 : *SYSBAS
			デジタル 署名入り
			プロファイル
			状態
			*USER

図 3-12 ANZOBJCVN プログラム変換の問題報告書



### 3.4.1 失われる属性

プログラム・オブジェクト用に生成される報告書では、変換時にプログラムが属性を失うことが示される場合もあります。図 3-13 に示されている例の「一部属性の逸失 (Some Lost Attributes)」列を見てください。

以下のオブジェクトは、属性の逸失にカウントされます。

- ▶ 「デジタル署名入り」の下に「YES」が表示されている場合、そのオブジェクトはデジタル署名を失います。この変更は、オブジェクトの妥当性検査時のみ問題になります。オブジェクトの機能には影響を与えません。
- ▶ 「プロファイル」の下に「YES」が表示されている場合、このオブジェクトはプロファイル作成による効果を失います。この変更はオブジェクトのパフォーマンスに影響を与えます。
- ▶ 状態が \*SYSTEM または \*INHERIT のオブジェクトは、\*USER 状態に低下します。この変更は、オブジェクトの機能に影響を与えることがあります。これらの状態はオペレーティング・システム用に予約されており、アプリケーションは使用してはなりません。

これらの属性についての詳細は、12 ページの 3.1.1、『用語の定義』を参照してください。

<b>オブジェクト変換の分析 - ライブラリー詳細報告書</b>		9/21/07 14:48:38	ページ 1
5722SS1 V5R4M0 060210 RCHAS10	収集の選択基準 - ANZOBJCVN OPTION(*COLLECT)		
ライブラリー	MILLSI		
ASPDEV	*		
SPLFILE	*YES		
オブジェクト	/		
サブツリー	*ALL		
ターゲット・リリース	V6R1M0		
報告書の選択基準 - ANZOBJCVN OPTION(*REPORT)			
報告書タイプ	*LIBDTL		
ライブラリー	MILLSI		
ASPDEV	*		
SPLFILE	*YES		
オブジェクト	/		
サブツリー	*ALL		
ターゲット・リリース	V6R1M0		
<b>オブジェクト変換の分析 - ライブラリー明細報告書</b>		9/21/07 14:48:38	ページ 2
5722SS1 V5R4M0 060210 RCHAS10			
プロセッサ・フィーチャー・コード . . . . .	7456		
補助記憶域プール (ASP) . . . . .	00001	ASP	資源名 : *SYSBAS
	----- カウンター -----	----- 徴候 -----	
	合計	変換が	属性の
ライブラリー	OBJ 数	必要	一部属性の
			逸失
			逸失
			変換
			時間
MILLSI	293	291	236
			55
			00:31:38

図 3-13 失われる属性を表示する ANZOBJCVN ライブラリー詳細報告書 (\*LIBDTL)

## 3.5 分析データに対する QUERY

ANZOBJCVN OPTION (\*COLLECT) コマンドによって収集されるデータは、QUSRSYS ライブラリー内の 4 つのファイルのいずれかに保管されます ( 詳細については、23 ページの 3.2.3、『データ収集に使用されるファイル』を参照してください)。各ファイルには、そのタイプの最後の収集からのデータのみが入っているのを知っておくことが重要です。つまり、後続の収集は、直前にこのファイルに収集されたデータを置き換えます。

必要に応じて、これらのファイルに対して QUERY を実行し、追加の報告書を生成できません。例えば、STRSQL コマンドを入力し、次に以下の QUERY を入力して、作成データのないプログラム・オブジェクトを抽出します。

```
SELECT DIOBLI, DIOBNM FROM QUSRSYS/QAIZACVN WHERE DICRTD=000
```

この QUERY では、プログラム作成データ域標識 (DICRTD) フィールドがゼロに設定されている場合 (これは、そのプログラム・オブジェクトは作成データを持たないことを示します)、ライブラリー名 (DIOBLI) とオブジェクト名 (DIOBNM) のフィールドが選択されます。

ANZOBJCVN コマンドによって作成されるファイルのフィールド定義については、63 ページの付録 B、『ファイルのフィールド定義の一覧表』を参照してください。

以下の QUERY は、ANZOBJCVN OPTION (\*COLLECT) コマンドで収集したデータの分析に役立つことがあります。

- ▶ 影響を受ける文字を含む名前を持つファイル・システム・オブジェクトを入手するための QUERY。  
SELECT QIZAOBJNAM FROM QUSRSYS/QAIZAOBJ WHERE QIZANAMFLG = 1  
Java プログラム変換を必要とするファイル・システム・オブジェクトを入手するための QUERY。  
SELECT QIZAOBJNAM FROM QUSRSYS/QAIZAOBJ WHERE QIZACVNFLG = 1
- ▶ Java プログラム変換を必要とするオブジェクトの絶対パス名、または影響を受ける Unicode 文字を含む名前を持つオブジェクトを入手するための QUERY。  
SELECT O.QIZAOBJNAM, D.QIZADIRNAM1 FROM QUSRSYS/QAIZAOBJ AS O INNER JOIN QUSRSYS/QAIZADIR AS D ON D.QIZADIRIDX = O.QIZADIRIDX WHERE O.QIZACVNFLG = 1 OR O.QIZANAMFLG = 1
- ▶ 変換時間が 1 秒を超えるスプール・ファイルの ASP を入手するための QUERY。  
SELECT SPASP,SPASPN FROM QUSRSYS/QAIZASPL WHERE SPTMCV > 1000000

**注：**上記の QUERY は、ジョブの CCSID が 65535 に設定されている場合、読めない出力を生成します。この問題を訂正するには、QCCSID システム値を 65535 以外の値（例えば、37）に変更することができます。あるいは、QUERY を以下のように変更することも可能です。

- ▶ 影響を受ける文字を含む名前を持つファイル・システム・オブジェクトを入手するための QUERY。  
SELECT CHAR(QIZAOBJNAM) FROM QUSRSYS/QAIZAOBJ WHERE QIZANAMFLG = 1
- ▶ Java プログラム変換を必要とするファイル・システム・オブジェクトを入手するための QUERY。  
SELECT CHAR(QIZAOBJNAM) FROM QUSRSYS/QAIZAOBJ WHERE QIZACVNFLG = 1
- ▶ Java プログラム変換を必要とするオブジェクトの絶対パス名、または影響を受ける Unicode 文字を含む名前を持つオブジェクトを入手するための QUERY。  
SELECT CHAR(O.QIZAOBJNAM), D.CHAR(QIZADIRNAM1) FROM QUSRSYS/QAIZAOBJ AS O INNER JOIN QUSRSYS/QAIZADIR AS D ON D.QIZADIRIDX = O.QIZADIRIDX WHERE O.QIZACVNFLG = 1 OR O.QIZANAMFLG = 1
- ▶ 変換時間が 1 秒を超えるスプール・ファイルの ASP を入手するための QUERY。  
SELECT CHAR(SPASP,SPASPN) FROM QUSRSYS/QAIZASPL WHERE SPTMCV > 1000000

## 3.6 アプリケーション開発者に関する考慮事項

他の人のためにアプリケーションを開発している場合、そのアプリケーションがお客様に販売するものであるか、社内の他の人のためにデプロイするものであるかに関係なく、プログラム変換について慎重に考慮してください。

### 3.6.1 アプリケーションを変換するか

最初に検討すべきことは、プログラムまたはサービス・プログラムが変換可能であるかどうかです。プログラムが十分な作成データを持っており、かつサポートされない方法で変更されていない場合、そのプログラムは変換可能です。一般に、V5R1 以降のリリースをターゲットとする MI プログラムは変換可能です。そのようなプログラムでは、その構成モジュールも V5R1 以降をターゲットとしていなければなりません。それ以前のリリースをターゲットとするプログラムも、そのプログラム識別情報の作成データ部分がまだ使用可能である限り、変換可能です。

**追加情報：**変換できないプログラムを調べる方法については、31 ページの 3.4、『変換されないプログラム』を参照してください。

前述のように、サポートされない方法でハードウェア命令ストリームを変更したプログラム、またはシステム状態で実行するように変更されたプログラムは、正常に変換することはできません。このようなプログラムは、サポートされるインターフェースだけを使用するように変更する必要があります。ユーザーの環境用のサポートされるインターフェースが存在しない場合は、Rich Diedrich [richd@us.ibm.com](mailto:richd@us.ibm.com) 宛に電子メールを送信して、新しいインターフェースを要求することができます。

何らかの理由で変換されないプログラムがある場合、V6R1 への移動を希望するお客様のために、ソフトウェアの 1 つ以上のバージョンを再コンパイルするかどうかを、お客様の設定に基づいて決める必要があります。プログラム変換の観点からお客様をサポートする方法を決めるのが最良です。例えば、アプリケーションを再コンパイルして、そのモジュールと

プログラムのすべてが V5R1 以降をターゲット・リリースにするようにすることができます。この新しくビルドされたアプリケーションは、お客様のシステム上で V6R1 で実行するように変換することができます。

代わりに、2つのバージョンのプログラムを提供するという選択も可能です。1つは V6R1 をターゲットとし、もう1つはそれ以前のリリースをターゲットとします。新規の V6R1 作成オプションや新しい言語機能を利用しない限り、同じソースを両方のバージョンで使用できます。アプリケーションのインストール・コードは、お客様が実行しているリリースに基づいて、正しいバージョンを選択してインストールできます。V6R1 上と以前のリリース上の両方でコンパイルすることができます。あるいは、アプリケーションを V6R1 上でのみコンパイルし、それを保管し、それを V5R4 または V5R3 に復元することも可能です。この場合は、復元時に変換が行われます。この V6R1 より前のバージョンも保管すれば、アプリケーションを1回だけコンパイルした後、変換を必要としない2つのバージョンが得られることとなります。

**V6R1 変換の支援**：V6R1 へのアプリケーションの変換に援助が必要な場合は、IBM ISV Enablement for i5/OS V6R1 にご連絡ください。氏名、役職、会社名、PartnerWorld 番号、および具体的な質問内容を E メールで [iaic@us.ibm.com](mailto:iaic@us.ibm.com) 宛にお送りください。この E メール・アドレスを使用して、V6R1 で実行する準備ができていないアプリケーションのバージョンを IBM に通知することもできます。

### 3.6.2 その他の決定基準

MI プログラムおよびサービス・プログラムが変換できることを確認した場合、変換時間は考慮要因となりますか。前述のように、ANZOBJCVN コマンドは特定のマシン・モデル上でプログラムを変換するために必要な時間の大きめ見積もりを提供します。例えば、プログラムの変換時間がインストール時にお客様の問題になるかどうかを考えてください。あるいは、アプリケーション内で呼び出しが実行される前にプログラムが変換される場合、意図せずにタイムアウト条件にかかってしまうようなプログラミングがなされていないか検討してください。

妥当な時間で正常に変換できる場合でも、新しいバージョンのアプリケーションの配布を検討する理由があります。例えば、アプリケーションでプロファイル作成を使用し、プロファイル作成データを削除してしまった場合、変換プロセスではプロファイル作成は適用されません。

考えられるもう1つの状態は、デジタル署名をプログラムに適用し、プログラムがお客様のシステムにロードされた後もそれを利用できるようにしたい場合です。デジタル署名はプログラム変換時に除去されます。これは、オブジェクトが変更されるためです。デジタル署名は、オブジェクトが変更されていないことと、予想された提供元から提供されたことを保証するために広く使用されています。このことは、デジタル署名はプログラムがロードされる前に最もよく使用されることを意味しています。しかし、アプリケーションによっては、デジタル署名を別の理由で使用する場合もあります。

アプリケーションの新しいバージョンを再コンパイルして再配布することに決めた場合、新規の作成オプションを活用することもできます。詳しくは、7 ページの 2.3.2、『パフォーマンス』を参照してください。新規の作成オプションは、アプリケーションのパフォーマンスを大幅に改善できる可能性があります。ただし、これは V6R1 をターゲットにしたプログラムでのみ利用できることに注意してください。

アプリケーションでプロファイル作成をまだ利用していなかった場合は、パフォーマンスを向上するためのもう1つの手段として、これを利用することを検討してください。アプリケーション・プロファイル作成は新しい機能ではありませんが、まだ十分に活用されていません。これを使用することにより、アプリケーションの全体的なパフォーマンスを最大 10% 改善できるという結果が得られています。詳細については、以下の Web アドレスで、

「System i ILE 概念 バージョン 6 リリース 1」(SD88-5033) 資料の『拡張最適化』の章を参照してください。

<http://publib.boulder.ibm.com/infocenter/systems/topic/books/sc415606.pdf>

### 3.6.3 適応コード生成

アプリケーションを再コンパイルして再配布するかどうかに関係なく、適応コード生成 (ACG) について検討してください。POWER6 テクノロジーをベースとするシステム上でアプリケーションを作成または変換すると、そのシステム上では他のシステムよりも高速でアプリケーションを実行できる場合があります。これは、アプリケーションが POWER6 プロセッサに固有のハードウェア・フィーチャーをどの程度幅広く使用しているかによって決まります。ただし、このアプリケーションを古いプロセッサが搭載されたシステムにロードする場合は、そのシステムが V6R1 を実行していても変換が必要になります。アプリケーション内のプログラムにとってロード時間が重要な場合は、ロード時間の差異をインストール説明書に記載する必要があります。

POWER6 テクノロジー固有のフィーチャーを使用してもアプリケーションのパフォーマンスがそれほど改善されない場合は、サポートされるすべてのプロセッサで使用可能なフィーチャーのみを使用するようにアプリケーションを作成するのが最良と思われます。この方法を使用すると、V6R1 を実行する POWER6 以外のシステムにプログラムをロードする場合、変換時間を回避できます。古いプロセッサを搭載したシステム上でプログラムを作成することもできますし、POWER6 プロセッサ上でプログラムを作成する際に ACG を「適用しない」こともできます。後者は、以下の方法のいずれかを使用して行えます。

- ▶ QIBM\_BN\_CREATE\_WITH\_COMMON\_CODEGEN 環境変数を 2 の値に設定する。
- ▶ アプリケーションの作成時に使用する CRT\* コマンドまたは CHG\* コマンドで、LICOPT('CodeGenTarget=Common') パラメーターを指定する。

一方、アプリケーションを作成または変換したシステムが V6R1 POWER6 システムではない場合、アプリケーションを最適化するために ACG を利用することができます。この場合、作成データが得られる場合は、V6R1 POWER6 システム上で以下のコマンドを実行します。

CHGPGM または CHGSRVPGM で FRCCRT(\*YES) を指定

プログラム作成データが得られない場合は、最初に QFRCCVNRST システム値を 7 に設定してから復元するか (例えば、保管ファイルから)、または RESTORE コマンドで FRCOBJCVN(\*YES \*ALL) パラメーターを指定します。

### 3.6.4 修正とテスト

変換はアプリケーションの処理全体に影響を与えるだけでなく、修正や更新を配布するためのプロセスにも影響を与えることがある点に注意してください。上にリストした各方面への影響を考慮してください。また、プログラム更新 (UPDPGM) コマンドは、更新されるプログラムの活動化も呼び出しも行わないので、プログラム変換をトリガーしない点にも注意してください。修正や更新の配布のために UPDPGM を使用する場合は、RESTORE コマンドで使用するパラメーターを変更して、変換を強制することも可能です。詳細については、55 ページの 4.3、『復元時のプログラム・オブジェクトの変換』を参照してください。

一般原則として、アプリケーションが V6R1 用に準備できているかどうかを確認するためのテスト方法は、他のリリースの場合と同じです。

## 3.7 その他のオブジェクト変換

MI プログラム・モデル・オブジェクトに加えてに、以下のオブジェクトが V6R1 へのアップグレード時に変換されます。

- ▶ Java プログラム
- ▶ 一部の統合ファイル・システムの名前
- ▶ スプール・ファイル

### 3.7.1 Java プログラム

Java プログラムは、ライブラリーではなくディレクトリー内にあります。i5/OS 統合ファイル・システム内で作成され、保管されているすべての Java プログラムを変換する必要がありますが、これは問題なく変換されるものと思われます。変換時間を見積もるには、Java プログラムを含んでいる統合ファイル・システム・ディレクトリーに対して ANZOBJCVN コマンドを実行できます。例えば、以下のような収集コマンドおよび報告コマンドを実行します。

```
ANZOBJCVN OPTION(*COLLECT) LIB(*NONE) SPLFILE(*NO) OBJ(Ð/millsiÐ)
ANZOBJCVN OPTION(*REPORT) RPTTYPE(*OBJSUM)
```

図 3-14 は、/millsi ディレクトリーについて生成されたオブジェクト要約報告書 (\*OBJSUM) の最初の 2 ページを示しています。

オブジェクト変換の分析 - オブジェクト要約報告書		9/21/07 14:56:54	ページ 1
5722SS1 V5R4M0 060210 RCHAS10			
収集の選択基準 - ANZOBJCVN OPTION(*COLLECT)			
ライブラリー	MILLSI		
ASPDEV	*		
SPLFILE	*YES		
オブジェクト	/millsi		
サブツリー	*ALL		
ターゲット・リリース	V6R1M0		
報告書の選択基準 - ANZOBJCVN OPTION(*REPORT)			
報告書タイプ	*OBJSUM		
ライブラリー	MILLSI		
ASPDEV	*		
SPLFILE	*YES		
オブジェクト	/millsi		
サブツリー	*ALL		
ターゲット・リリース	V6R1M0		
プロセッサ・フィーチャー・コード		7456	
収集された日付		09/21/07	
変換が必要なオブジェクト数		33	
ディレクトリー・オブジェクトの推定変換時間合計		00:02:29	
オブジェクト変換の分析 - オブジェクト要約報告書		9/21/07 14:56:54	ページ 2
5722SS1 V5R4M0 060210 RCHAS10			
ディレクトリー :/TOMARS2			
--- カウンター --- ----- 徴候 -----			
変換が 必要	ユニコード 文字	変換 時間	
0	0	00:00:00	
変換 推定時間 (秒)	オブジェクト		
-----			
このセクションで収集されたデータはありません。			
ユニコード文字			
-----			
このセクションで収集されたデータはありません。			

図 3-14 /millsi ディレクトリーに対する ANZOBJCVN オブジェクト要約報告書 (\*OBJSUM)



図 3-15 は、/millsi ディレクトリーのオブジェクト詳細報告書 (\*OBJDTL) の最初の 2 ページを示しています。Java プログラムは、ライブラリー内のプログラム・オブジェクトの場合のように、変換のために MI 作成データに依存することはありません。また、Java プログラム・オブジェクトの変換時間は、ほとんどの場合 1 秒未満と考えられます (すべての場合ではないとしても)。

```

オブジェクト変換の分析 - オブジェクト詳細報告書
5722SS1 V5R4M0 060210 RCHAS10
9/21/07 15:01:06
                                     ページ 1

収集の選択基準 ANZOBJCVN OPTION(*COLLECT)
ライブラリー      MILLSI
ASPDEV            *
SPLFILE          *YES
オブジェクト      /millsi
サブツリー        *ALL
ターゲット・リリース V6R1M0

報告書の選択基準 - ANZOBJCVN OPTION(*REPORT)
報告書タイプ      *OBJDTL
ライブラリー      MILLSI
ASPDEV            *
SPLFILE          *YES
オブジェクト      /millsi
サブツリー        *ALL
ターゲット・リリース V6R1M0

オブジェクト変換の分析 - オブジェクト詳細報告書
5722SS1 V5R4M0 060210 RCHAS10
9/21/07 15:01:06
                                     ページ 2

プロセッサ・フィーチャー・コード . . . . . : 7456

ディレクトリー :/TOMARS2
---- カウンター ----      徴候 ----
      変換が      ユニコード      変換
      必要      文字      時間
-----
                      0          0          00:00:00

変換      オブジェクト
推定時間
(秒)
-----
このセクションで収集されたデータはありません。

ユニコード文字
-----
このセクションで収集されたデータはありません。

```

図 3-15 /millsi ディレクトリーの ANZOBJCVN オブジェクト詳細報告書 (\*OBJDTL)

### 3.7.2 統合ファイル・システムの名前

操作するオブジェクトを識別するために、統合ファイル・システム・コマンドまたは API で名前が指定されると、名前の検索が実行されます。大 / 小文字の区別をしないファイル・システム内の名前の検索は、定義される Unicode 文字とそのリリースで有効な大 / 小文字規則による影響を受けます。追加の文字が定義され、その新しい文字をサポートするために大 / 小文字規則が更新された場合、この変更による影響を受ける文字を含んでいる名前は、検出されなくなる可能性があります。

i5/OS 統合ファイル・システムはすべての名前を Unicode で保管しており、現在は Unicode Standard 2.0 をサポートしています。V6R1 では自動変換が実行されて、大 / 小文字の区別をしないファイル・システム内の統合ファイル・システム・ディレクトリーは、Unicode Standard 4.0 をサポートするように更新されます。この変換に含まれる大 / 小文字の区別をしないファイル・システムは、「ルート」(/) と、CASE(\*MONO) で作成されたユーザー定義ファイル・システム (任意の ASP 内の) です。

ANZOBJCVN ツールは、V6R1 の Unicode の変更によって影響を受ける、大 / 小文字の区別をしないファイル・システム内の統合ファイル・システム・オブジェクト名を識別します。あるオブジェクト名が影響を受けると識別された場合、これは V6R1 で自動名前変換が実行されたときに、そのオブジェクトまたは同じ親ディレクトリーにリンクされた別のオブジェクトのいずれかが名前変更される可能性があることを意味します。したがって、1 つだけのオブジェクトが名前変更された場合でも、複数のオブジェクトが影響を受けるとして識別されることがあります。影響を受けるオブジェクトのどれが名前変更されるオブジェクトであるかは保証されません。

大 / 小文字の区別をしないファイル・システムでは、システムは名前の検索時に大文字と小文字を区別しません。大 / 小文字規則の変更は、大文字と小文字を区別していない場合、どの文字を同一と見なすかに影響を与えます。このため、自動名前変換時には、Unicode Standard 2.0 では同じと見なされていなかった名前が、Unicode Standard 4.0 では同じと見なされるようになる可能性があります。オブジェクトの 1 つを名前変更する必要があります。この状態が生じた場合、どのオブジェクトが名前変更されるかは予測不能です。この状態が起きたときにどのオブジェクトが名前変更されるかは、ディレクトリー内のリンクの順序や、オブジェクトの 1 つが損傷しているかどうかなど、さまざまな要因が影響を与える可能性があります。

ANZOBJCVN コマンドで、Unicode 変換による影響を受けるとして識別されたオブジェクトに対して、何をする必要がありますか。自動名前変換が実行されるのを待つのではなく、V6R1 をインストールする前に、影響を受けるオブジェクトの名前を変更することができます。このようにすると、どのような名前変更に対しても、より制御された方法で対処することができます。オブジェクト名を変更し、再度 ANZOBJCVN コマンドを実行して、起きる可能性のある名前の問題が解決されたことを確認することができます。

**名前変更されるオブジェクト :** ANZOBJCVN コマンドでは Unicode の名前変換による影響を受けるオブジェクトが識別されなかった場合でも、Unicode Standard の変更が原因で、あるいは V6R1 へのアップグレード前にシステム上で何らかの名前変更が行われた場合には、自動名前変換によって一部のオブジェクトが名前変更される可能性があります。

追加情報については、以下の Web アドレスで、「Informational APAR II14306」を参照してください。

[http://www-912.ibm.com/n\\_dir/nas4apar.nsf/c79815e083182fec862564c00079d117/3af47a966c4df94586257306003c6868?OpenDocument&Highlight=2,ii14306](http://www-912.ibm.com/n_dir/nas4apar.nsf/c79815e083182fec862564c00079d117/3af47a966c4df94586257306003c6868?OpenDocument&Highlight=2,ii14306)

### 3.7.3 スプール・ファイル

デフォルトでは、スプール・ファイルの変換は、システムの IPL 時または ASP をオンに構成変更したときに実行されます。スプール・ファイル・データは、ANZOBJCVN コマンドによって収集され、システムのアップグレードを計画する際に役立つ情報を提供します。見積もり時間は、SRC C900 2AAA のインストール IPL 時または ASP グループのオンに構成変更するステップ C900 29B0 時に、システムがスプール・ファイルを変換するためにかかる予想時間です。

情報を収集し、報告書を生成した後、スプール・ファイル変換時間の詳細が、図 3-16 の例に示すように報告書に表示されます。

変換されるスプール・ファイルの合計 . . . . .	697
スプール・ファイルの推定変換時間合計 . . . . .	00:00:06

図 3-16 スプール・ファイルの変換見積もり時間

収集されたデータのスプール・ファイル数は、QSPLxxxx ライブラリー内のすべてのデータベース・メンバーを表します。ここで、xxxx はブランク、もしくは ASP グループの場合は 0002-0032 または 0033-0255 です。

スプール記憶域再利用 (RCLSPLSTG) DAYS(\*NONE) コマンドを使用して、SYSBAS の QSPLxxxx ライブラリー内のデータベース・メンバーの数を削減することは、一般に推奨されません。ASP グループの QSPLxxxx ライブラリー内の使用可能なデータベース・メンバーの数を減らすと、スプール・ファイルの作成時に重大なパフォーマンス問題を引き起こし、その結果、システムまたはアプリケーションが不定期間、フル容量に達して使用不能になる可能性があります。以下の Web アドレスで、「スプール・ファイル記憶域再利用の体験レポート (Reclaim SpooledFile Storage Experience Report)」を参照してください。

<http://publib.boulder.ibm.com/infocenter/iserics/v5r4/topic/experience/spoolreclaim.pdf>

データ域を使って変換を遅らせることができます。ただし、スプール・ファイルの変換を据え置くと、変換が完了するまで、既存のスプール・ファイルが利用できなくなることに注意してください。スプール・ファイルの変換を遅らせることについての詳細は、以下の Web アドレスで、「i5/OS および関連ソフトウェアのインストール、アップグレードおよび削除 (SD88-5002-10)」を参照してください。

<http://publib.boulder.ibm.com/infocenter/systems/topic/rzahc/sc415120.pdf>

スプール・ファイル変換の追加情報については、以下の Web アドレスで、「Informational APAR II14306」を参照してください。

[http://www-912.ibm.com/n\\_dir/nas4apar.nsf/c79815e083182fec862564c00079d117/3af47a966c4df94586257306003c6868?OpenDocument&Highlight=2,ii14306](http://www-912.ibm.com/n_dir/nas4apar.nsf/c79815e083182fec862564c00079d117/3af47a966c4df94586257306003c6868?OpenDocument&Highlight=2,ii14306)

**ヒント:** オペレーティング・システムをアップグレードする前に、システム上に保持されているスプール・ファイルの数を減らしてください。多数のスプール・ファイルが出力キュー上に残っていると、変換時間が長くなります。定期的なシステム保守の一部としてスプール・ファイルを除去または保存することも検討する必要がありますが、これは特にシステム・アップグレードを実行する前に必要です。

## 3.8 サポートされないプロダクトの削除

V6R1 では、プログラムを削除する前に、削除対象のプログラムを変換する必要はありません。しかし、ある特定のプロダクトが削除操作時に出口プログラムを使用し、その出口プログラムが変換できない何らかの別のプログラムを呼び出す場合は、V6R1 上でライセンス・プログラム削除 (DLTLICPGM) コマンドを正常に完了できない可能性があります。この問題は、V6R1 にアップグレードする前に、サポートされないプロダクトを削除することで回避できます。ただし、後に V6R1 より前のリリースで必要になる場合は、削除する前に古いプロダクトを保存しておくことができます。



## プログラム変換方式

この章では、V6R1 で実行するためのプログラムの変換のさまざまな方式について説明します。

**重要：**この章で説明する変換方式は、V6R1 にアップグレード済みまたはマイグレーション済みのシステム上で実行する必要があります。

i5/OS V6R1 のアップグレードまたはマイグレーション・プロセスを開始する前に、オブジェクト変換のための準備をしておくことをお勧めします。変換されないために V6R1 上では使用できないプログラム・オブジェクトが見つかることがあります。オブジェクト変換分析 (ANZOBJCVN) コマンドの使用法についての詳細は、11 ページの第 3 章、『変換の準備』を参照してください。このコマンドは、V6R1 へのアップグレードまたはマイグレーションを開始する前に必要な処置を判別するのに役立ちます。

ANZOBJCVN コマンドは、V6R1 上では使用できません。ユーザーが V6R1 システム上で、変換が必要なオブジェクトを確認する必要がある場合は、\*CHECK オプションを指定したオブジェクト変換開始 (STROBJCVN) コマンドを使用する必要があります。詳しくは、48 ページの 4.2.2、『STROBJCVN コマンドのパラメーター』を参照してください。

**重要：**V6R1 システム上でいかなる種類の変換も開始する前に、PTF MF44237 がインストール済みであることを確認してください。この PTF は、プログラム変換中にまれに発生するシステム・ハングを防止します。

## 4.1 プログラム変換の概要

この段階では、変換されないプログラム・オブジェクトが確認されており、これらのオブジェクトの再コンパイルまたは V6R1 互換バージョンの入手が計画済みであるものと想定しています。

V6R1 にアップグレードする際には、インストール・メディアによって、さまざまな IBM 提供プログラム・オブジェクトが V6R1 互換形式で提供されます。こうしたプログラム・オブジェクトは、変換の必要はありません。ただし、一部の IBM 提供プログラム (例えば、V6R1 のスキップ・シップ・プロダクト) は、変換が必要です。

システムを V6R1 にアップグレードした後で、プログラム変換を実行できます。変換の必要があるオブジェクトのリストは (ある場合)、2 つの部分から構成されます。

- ▶ 11 ページの第 3 章、『変換の準備』で説明されたプロセスで ANZOBJCVN コマンドによって識別されたオブジェクト  
これには、ユーザー・ライブラリー内のプログラム・オブジェクトと、ディレクトリー内のファイルに付加された Java プログラムが含まれます。
- ▶ V6R1 のスキップ・シップとして識別されている、IBM ライセンス・プログラム・プロダクト (LPP)  
これは、特定の V6R1 バージョンがないプロダクトです。これらのプロダクトのリスト、およびプロダクトがインストールされているライブラリーについては、71 ページの付録 C、『変換を必要とする IBM ソフトウェア・スキップ・シップ・プロダクト』を参照してください。

V6R1 互換形式への変換は、一回限りの操作です。変換を必要とするプログラム・モデル・オブジェクトの場合、選択された変換方式によっては、変換が実行されている間、アプリケーションが利用できなくなることがあります。

**変更日とオブジェクト・サイズ:** V6R1 互換形式に変換されるプログラム・モデル・オブジェクトの場合、変更日時は変換の時刻に変更され、オブジェクト・サイズも変更されることがあります。これらの変更は、監査に影響を与えることがあります。

変更済みオブジェクト保管 (SAVCHGOBJ) コマンドを使用する場合、どのオブジェクトが保管されるかは、変換の影響を受けることに注意してください。

プログラム・モデル・オブジェクトのサイズ変更は小さいものと予想されます。特定のシステム上のすべてのプログラム・モデル・オブジェクトを対象にして、1 桁のパーセントの増加というのが最も可能性の高い値です。また、プログラム・モデル・オブジェクトが消費する記憶域の量は、ほとんどの場合、全体の記憶域使用量のごく一部にすぎません。したがって、プログラム変換の結果としての記憶域の増加は、あまり重要ではないはずですが、小さい非常に古いプログラムは増加率が大きくなる傾向がありますが、それでも消費する記憶域の量は極めて小さいものです。

プログラム・オブジェクト変換を実行するには、以下の方式を使用できます。

- ▶ V6R1 STROBJCVN CL コマンドを使用して、手動でライブラリー内のプログラム・オブジェクトに対する変換を開始する  
詳細については、47 ページの 4.2、『STROBJCVN を使用した変換』を参照してください。
- ▶ 復元時  
詳細については、55 ページの 4.3、『復元時のプログラム・オブジェクトの変換』を参照してください。

▶ 最初のプログラム呼び出し時

詳細については、57 ページの 4.4、『最初のプログラムの実行時または呼び出し時の変換』を参照してください。

**STROBJCVN コマンドを使用することを推奨：**すでにシステム上にあるオブジェクトの場合は、「最初の呼び出し」による変換を待つよりもむしろ、STROBJCVN コマンドを使用してください。プログラムを呼び出す前に変換プロセスを開始すると、マシン資源の使用制御に関して最大の柔軟性が得られます。また、独自のタイミング制約に依存するアプリケーションや最初の呼び出しが複数のユーザーによって同時に実行されるアプリケーションの場合、実行時動作に影響を与える可能性も回避できます。

## 4.2 STROBJCVN を使用した変換

V6R1 STROBJCVN コマンドは、手動でオブジェクト変換を開始するために提供されています。オブジェクトは、旧レベルのオペレーティング・システムで使用されていた形式から、i5/OS V6R1 上で使用するために必要な形式に変換されます。

**重要：**STROBJCVN コマンドは i5/OS の以前のリリースにも存在しますが、V6R1 互換形式への変換をサポートするのは、V6R1 バージョンのコマンドだけです。

STROBJCVN コマンドでは、特定のオブジェクトについて変換が必要かどうかを検査するオプションが提供されます。ただし、この検査オプションを V6R1 上実行することは、V5R3 または V5R4 上で ANZOBJCVN コマンドを実行することの良い代替策ではありません。

STROBJCVN コマンドは、新規の Unicode 文字と大 / 小文字規則によって影響を受けるディレクトリーを変換しません。この変換は、オペレーティング・システムによって自動的に行われます。詳しくは、42 ページの 3.7.2、『統合ファイル・システムの名前』を参照してください。

STROBJCVN コマンドは、スプール・ファイルも変換しません。変換はオペレーティング・システムによって自動的に行われます。詳しくは、43 ページの 3.7.3、『スプール・ファイル』を参照してください。

STROBJCVN コマンドは、ANZOBJCVN 分析に含まれていない他のタイプのオブジェクト (ライブラリー内のファイルなど) を検査または変換することができます。ANZOBJCVN で分析されるオブジェクト・タイプのリストは、12 ページの 3.1、『オブジェクトの分析』を参照してください。

### 4.2.1 STROBJCVN の使用上の注意

以下の注意事項は、STROBJCVN CL コマンドの使用に関するものです。

- ▶ このコマンドを実行するには、全オブジェクト (\*ALLOBJ) 特殊権限が必要です。
- ▶ このコマンドを実行する前に、このコマンドによって参照されるすべての 1 次または 2 次補助記憶域プール (ASP 33-255) がオンに構成変更され、状況が「*Available*」になっていることが必要です。
- ▶ このコマンドを実行する前に、このコマンドによって参照されるすべてのユーザー定義ファイル・システムがマウントされている必要があります。
- ▶ 「読み取り専用」のユーザー定義ファイル・システム内にあるオブジェクトは変換されません。

**重要:** オプション \*CONVERT を指定したこのコマンドは実行に長時間かかることがあります。すべてのシステムについて、特に対話式 (5250 オンライン・トランザクション処理 (OLTP)) 機能が限定されているシステムでは、このコマンドをバッチ・ジョブとして実行することをお勧めします。

また、\*ALLUSR でこのコマンドを使用する前に、最大プール・ファイル・システム値 (QMAXSPLF) を最大値に変更することをお勧めします。SQL 変換は、SQL を含むオブジェクトを変換するたびにプール・ファイルの作成と削除が必要になるため、多数の SQL オブジェクトを持つシステムでは 9999 カウントに達してしまう可能性があります。

## マルチスレッド化

STROBJCVN コマンドは、プログラム、サービス・プログラム、およびモジュールを変換するために複数のスレッドを使用する場合があります。デフォルトでは、スレッドの数は利用可能なプロセッサの数に基づいて計算されます。現在は、利用可能な 1 プロセッサにつき 2 つのスレッドが使用されています。データ域を使用して、このデフォルトのスレッド使用を指定変更することができます。これを使用して、同時に使用されるスレッドの数を多くしたり、少なくしたりすることができます。

データ域を作成するには、次のようなコマンドを使用します。

```
CRTDTAARA DTAARA(QGPL/QIZANBRTHD) TYPE(*DEC) LEN(10 0) VALUE(x) TEXT(ⒹNumber of threads for STROBJCVND)
```

データ域を使用する場合、これは STROBJCVN コマンドのすべてのインスタンスに適用されます。ジョブごとの制御メカニズムはありません。

ほとんどのシステムでは、プロセッサあたり 2 スレッドというデフォルト値で妥当です。スレッドの数を変更するためのオプションは、STROBJCVN コマンドの動作をシステム要件に合わせて調整したいユーザー向けに提供されています。

**ヒント:** システム活動処理 (WRKSYSACT) コマンドは、Performance Tools (5761-SS1) プロダクトの一部に含まれており、アクティブ・スレッドをモニターするための容易な手段として利用できます。

## 4.2.2 STROBJCVN コマンドのパラメーター

STROBJCVN コマンドの使用時に、\*CHECK または \*CONVERT のいずれかを入力するように求めるプロンプトが出されます。

- ▶ \*CHECK を指定すると、指定されたライブラリーまたはディレクトリー内のオブジェクトの検査を開始し、それらのオブジェクトがすでに変換済みであるか、あるいはまだ変換が必要かどうかを判別します。\*CHECK パラメーターは、プログラム・オブジェクトの変換が可能かどうかについてのフィードバックも戻します。詳細については、51 ページの 4.2.3、『STROBJCVN OPTION(\*CHECK)』を参照してください。
- ▶ \*CONVERT を指定すると、指定されたライブラリーまたはディレクトリー内の該当するすべてのオブジェクトに対して変換プロセスを開始します。詳細については、53 ページの 4.2.4、『STROBJCVN OPTION(\*CONVERT)』を参照してください。

さらに、STROBJCVN コマンドには、以下のパラメーターが含まれています。

- ▶ ライブラリー (LIB)  
このパラメーターは、オブジェクトの検査を実行する、または V6R1 で使用するために必要な形式への変換を実行する、ライブラリーを指定します。
  - \*ALLUSR



すべてのユーザー・ライブラリーが選択されます。文字 Q で始まっていない名前を持つすべてのライブラリーが選択されます。ただし、18 ページの 3.2.1、『ANZOBJCVN \*COLLECT コマンドのパラメーター』に詳述されているライブラリーは除きます。

– \*NONE

ライブラリーは選択されません。LIB(\*NONE) は、ディレクトリーにあるオブジェクトだけを変換または検査したい場合に指定してください。

– ライブラリー名

オブジェクトを検査または変換するライブラリーの名前を指定します。

▶ オブジェクト・タイプ (OBJTYPE)

このパラメーターは、ライブラリー内の変換する必要があるオブジェクト・タイプを指定します。オプション (OPTION) パラメーターに \*CHECK が指定されている場合、オブジェクト・タイプ・パラメーターは無視されます。ライブラリー (LIB) パラメーターに \*NONE が指定されている場合、オブジェクト・タイプ・パラメーターは無視されます。

– \*ALL

指定されたライブラリー内のすべての (\*PGM) オブジェクト、サービス・プログラム (\*SRVPGM) オブジェクト、モジュール (\*MODULE) オブジェクト、およびデータベース・ファイル (\*FILE) オブジェクトが変換されます。さらに、SQL ステートメントを含むすべての \*PGM、\*SRVPGM、および \*SQLPKG オブジェクト内、および外部ストアード・プロシージャーを実装するために使用される \*PGM および \*SRVPGM オブジェクトに内に保管された SQL 情報も変換されます。

– \*FILE

指定されたライブラリー内のデータベース・ファイル・メンバー・オブジェクトだけが変換されます。この変換の簡単な説明は、53 ページの 4.2.4、『STROBJCVN OPTION(\*CONVERT)』を参照してください。

– \*ALLPGM

指定されたライブラリー内のすべてのプログラム (\*PGM) オブジェクトおよびサービス・プログラム (\*SRVPGM) オブジェクトが変換されます。

– \*SQL

SQL ステートメントを含むすべての \*PGM、\*SRVPGM、および \*SQLPKG オブジェクト内、および外部ストアード・プロシージャーを実装するために使用される \*PGM および \*SRVPGM オブジェクト内に保管された SQL 情報も変換されます。

▶ ASP 装置 (ASPDEV)

このパラメーターは、検査または変換するライブラリー・オブジェクト用の記憶域が割り振られている ASP 装置を指定します。ライブラリーを格納する ASP が、スレッドのライブラリー・ネーム・スペースの一部に含まれていない場合は、必ずこのパラメーターを指定して、正しいライブラリーが使用されるようにする必要があります。

**オブジェクト・パラメーター:** ASP 装置パラメーターは、オブジェクト (OBJ) パラメーター (後述) で指定されたオブジェクトには適用されません。独立 ASP 名がオブジェクトのパス名の一部として含まれているからです。

– \*(アスタリスク)

ライブラリーを見つけるために、現在スレッドのライブラリー・ネーム・スペースの一部として含まれている ASP が検索されます。これには、システム ASP (ASP 1)、すべての定義済み基本ユーザー ASP (ASP 2-32)、および、スレッドに ASP グループがある場合は、スレッドの ASP グループ内の 1 次および 2 次 ASP が含まれます。

- \*ALLAVL

すべての使用可能な ASP が検索されます。これには、システム ASP (ASP1)、すべての定義済み基本ユーザー ASP (ASP 2-32)、およびすべての使用可能な 1 次および 2 次 ASP (ASP 33-255 のうち状況が「Available」のもの) が含まれます。

- \*CURASPGRP

スレッドに ASP グループがある場合、スレッドの ASP グループ内の 1 次および 2 次 ASP を検索して、ライブラリーを見つけます。システム ASP (ASP 1) と定義済み基本ユーザー ASP (ASP 2-32) は検索されません。スレッドに ASP グループが関連付けられていない場合、エラーが出されます。

- \*SYSBAS

ライブラリーを見つけるために、システム ASP (ASP 1) とすべての定義済み基本ユーザー ASP (ASP 2-32) が検索されます。スレッドに ASP グループがある場合でも、1 次または 2 次 ASP は検索されません。

- 名前

このオプションを使用して、検索対象の 1 次または 2 次 ASP 装置の名前を指定することができます。1 次または 2 次 ASP は、活動状態であり (ASP 装置をオンに構成変更することによって活動化します)、状況が「Available」でなければなりません。システム ASP (ASP 1) と定義済み基本ユーザー ASP (ASP 2-32) は検索されません。

▶ オブジェクト (OBJ)

このパラメーターは、検査または変換される、Java プログラムが付加されたファイルのパス名、または名前に突き合わせるパターンを指定します。オブジェクトのパス名は、単純名でも、オブジェクトが置かれているディレクトリー名で修飾された名前でも構いません。パターンは、パス名の最後の部分で指定できます。アスタリスク (\*) は任意の文字数に突き合わせ、疑問符 (?) は単一文字に突き合わせます。パス名が修飾されているか、パターンを含んでいる場合、パス名をアポストロフィで囲む必要があります。ルート・ファイル・システム内の該当するオブジェクトを検査または変換するには、スラッシュ (/) を指定する必要があります。

\*NONE を指定すると、どの統合ファイル Java プログラムも検査または変換されないことを意味します。

▶ ディレクトリー・サブツリー (SUBTREE)

このパラメーターは、検査機能または変換機能が実行されるサブツリーを決めます。

**統合ファイル・システム・オブジェクト:** このパラメーターは、統合ファイル・システム内のオブジェクトにのみ適用されます。これは、OBJ パラメーターが指定されている場合にのみ有効です。

\*ALL が使用されている場合、オブジェクト (OBJ) パラメーターで指定された該当のオブジェクトが検査または変換されます。オブジェクトがディレクトリーの場合、その内容、およびそのサブディレクトリーのすべての内容が検査または変換されます。

コマンドが特定のディレクトリーのサブツリーの処理を開始すると、その後に検出されて処理されるオブジェクトは、指定のディレクトリー・ツリー内のオブジェクトの編成を更新する操作によって影響を受けることがあります。これには以下の操作が含まれます (ただし、これだけに限りません)。

- オブジェクト・リンクの追加、削除、または名前変更
- ファイル・システムのマウントまたはアンマウント
- コマンドを呼び出すプロセスの有効ルート・ディレクトリーの更新
- シンボリック・リンクの内容の更新

ディレクトリーのサブツリーを処理するために、システム・コードは、処理中にジョブ内で開くことができるファイル記述子の最大数を増やすことがあります。これは、記述子の不足が原因でコマンドが失敗することがないようにするために行われます。このジョブ有効範囲内の最大値は、コマンドが完了してもリセットされません。

\*NONE が使用されている場合、オブジェクト (OBJ) パラメーターで指定された該当のオブジェクトが検査または変換されます。オブジェクトがディレクトリーの場合、ディレクトリーは必要に応じて検査または変換されますが、その内容は検査または変換されません。

### 4.2.3 STROBJCVN OPTION(\*CHECK)

\*CHECK オプションを使用して STROBJCVN コマンドを実行した場合、LIB パラメーターおよび OBJ パラメーターで位置指定されたオブジェクトが検査され、変換が必要かどうかは判別されます。このコマンドでは、ライブラリー内の以下のタイプのプログラム・モデル・オブジェクトが検査されます。

- ▶ プログラム (\*PGM)
- ▶ サービス・プログラム (\*SRVPGM)
- ▶ モジュール (\*MODULE)

統合ファイル・システムでは、i5/OS 上で実行するために作成された Java プログラムを含むストリーム・ファイルが、このコマンドによって検査されます。「ルート」(/)、QOpenSys、およびユーザー定義ファイル・システム (UDFS) 内のストリーム・ファイルのみが含まれます。

図 4-1 は、STROBJCVN OPTION(\*CHECK) コマンド・パラメーターを示しています。この例では、コマンドは \*ALLUSR ライブラリー内のタイプ \*PGM、\*SRVPGM、または \*MODULE のすべてのオブジェクトを検査しますが、統合ファイル・システム内の Java プログラムは検査しません。STROBJCVN OPTION(\*CHECK) コマンドは、すでに変換済みのオブジェクトの数、まだ変換が必要なオブジェクトの数、および変換できないオブジェクトを報告します。

オブジェクト変換開始 (STROBJCVN)

選択項目を入力して、実行キーを押してください。

オプション . . . . .	> *CHECK	*CONVERT, *CHECK
ライブラリー . . . . .	> *ALLUSR	名前、総称 *, *ALLUSR...
オブジェクト・タイプ . . . . .	*ALL	*ALL, *FILE, *ALLPGM, *SQL
ASP 装置 . . . . .	*	名前、*, *ALLAVL...
オブジェクト . . . . .	*NONE	
ディレクトリー・サブツリー . . . . .	*ALL	*ALL, *NONE

終り

F3= 終了    F4= プロンプト    F5= 最新表示    F12= 取り消し    F13= この画面の使用法  
F24= キーの続き

図 4-1 STROBJCVNOPTION(\*CHECK) コマンド

\*CHECK プロセスの結果を見るには、セッション・ジョブ・ログを参照してください。  
 図 4-2 は、変換されなかったオブジェクトに関するジョブ・ログからのメッセージの詳細を示しています。

```

CPIB0EA Information 10 06/25/07 09:13:19.272366 QIZAECVN
QSYS 0975 QUIMNDRV QSYS 060F
Message . . . . : Object LIBHT451 has not been converted.
Cause . . . . : Start Object Conversion(STROBJCVN) OPTION(*CHECK) found
object LIBHT451 of type *SRVPGM in library CMC1 that has not been converted
yet to the current release of the operating system. Recovery . . . :
To convert the object to the current release of the operating system, run
STROBJCVN OPTION(*CONVERT) LIB(CMC1).
  
```

図 4-2 STROBJCVN OPTION(\*CHECK) の実行後のジョブ・ログ内のメッセージの例

コマンドは、メッセージ CPCB0D5 で完了するはずですが、これは、検査の結果を要約しています。図 4-3 は、このメッセージの例を示しています。

```

Additional Message Information

Message ID . . . . : CPCB0D5      Severity . . . . : 00
Message type . . . . : Completion
Date sent . . . . : 11/01/07     Time sent . . . . : 08:58:05

Message . . . . : Checking of objects was successful.
Cause . . . . : Start Object Conversion(STROBJCVN) OPTION(*CHECK) ended
with the following results.
  7 objects have not been converted yet. 159 objects did not need to be
converted. 166 objects were processed in *ALLUSR.
  0 objects have not been converted yet. 0 objects did not need to be
converted. 0 objects were processed in *N.
  Objects did not need to be converted due to one of the following reasons:
  - Some objects were already converted.
  - Some objects do not have all the creation data.
  If *N or 0 appears for any fields, then this information does not apply.

Bottom

Press Enter to continue.

F3=Exit  F6=Print  F9=Display message details
F10=Display messages in job log  F12=Cancel  F21=Select assistance level
  
```

図 4-3 メッセージ CPCB0D5 の例

「0 objects have not been converted yet.」で始まる追加メッセージ情報の部分に注意してください。このセクションは、その前にあるライブラリー情報と重複しているように見えますが、値が入力されていません。これは重複しているのではなく、ディレクトリー内のオブジェクトを参照しています。このコマンド呼び出しでは、オブジェクト・パラメーターが \*NONE に設定されていたので、ディレクトリー内のオブジェクトは検査されませんでした。

このコマンドで出されたメッセージのプール・ファイルを生成するための出力パラメーターはありません。しかし、次のようなジョブ・ログ表示 (DSPJOBLOG) コマンドを使用して、セッション・ジョブ・ログのプール・ファイルを生成することで、情報を入手できます。

```
DSPJOBLOG OUTPUT(*PRINT)
```

ヒント:メッセージを画面上で直接見たい場合は、コマンド入力画面 (CALL QCMD) から、次のコマンドを入力してください。

```
STROBJCVN OPTION(*CHECK)
```

#### 4.2.4 STROBJCVN OPTION(\*CONVERT)

\*CONVERT オプションを使用して STROBJCVN コマンドを実行すると、オブジェクト変換プロセスを開始します。変換の必要がないプログラム・モデル・オブジェクトは変換されません。このコマンドでは、ライブラリー内の以下のタイプのプログラム・モデル・オブジェクトが、必要に応じて変換されます。

- ▶ プログラム (\*PGM)
- ▶ サービス・プログラム (\*SRVPGM)
- ▶ モジュール (\*MODULE)

SQL データも、このコマンドによって変換されます。プログラム・オブジェクト、サービス・プログラム・オブジェクト、および SQL パッケージ (\*SQLPKG) オブジェクトは、外部ストアド・プロシージャのために使用される SQL ステートメントとデータを含んでいることがあります。プログラム・オブジェクト内に含まれている SQL データの変換は、プログラム自体の変換とは別に処理されます。

統合ファイル・システムでは、i5/OS 上で実行するために作成された Java プログラムを含むストリーム・ファイルも、このコマンドによって変換されます。「ルート」(/)、QOpenSys、およびユーザー定義ファイル・システム内のストリーム・ファイルのみが含まれます。

**重要:** STROBJCVN OPTION(\*CONVERT) は、ライブラリー内のさまざまなタイプのファイル・オブジェクトも処理します。これらのファイルが V5R4 用に変換されていない場合 (V5R3 から V6R1 にアップグレードした場合など)、これらのファイル・オブジェクトもある種の変換が行われます。

STROBJCVN OPTION(\*CONVERT) は、ファイル変換が必要ときには常にプロンプトを出すアクションを取る必要があります。ファイル変換がすでに行われている場合でも (V5R4 からアップグレードする場合はその可能性あります)、多数のファイルがあるライブラリーでは、変換が行われなくてもかかわらず、STROBJCVN OPTION(\*CONVERT) の実行時間が長くなる場合があります。ファイルの処理時間は ANZOBJCVN の見積もり時間に含まれていないことに注意することが重要です。

ライブラリーに相当数のファイルがある場合、特に V5R4 からアップグレードする場合には、\*ALLUSR ライブラリーを対象とする代わりに、個々のライブラリーを対象として STROBJCVN (\*CONVERT) を実行することをお勧めします。

図 4-4 は、\*CONVERT オプションを使用した STROBJCVN コマンドの例を示しています。この場合、コマンドは \*ALLUSR ライブラリー内のオブジェクトの変換を開始しますが、統合ファイル・システム内のファイルに接続された Java プログラムは変換しません。

オブジェクト変換開始 (STROBJCVN)		
選択項目を入力して、実行キーを押してください。		
オプション . . . . .	*CONVERT	*CONVERT, *CHECK
ライブラリー . . . . .	*ALLUSR	名前、*ALLUSR,*NONE
オブジェクト・タイプ . . . . .	*ALL	*ALL, *FILE, *ALLPGM, *SQL
ASP 装置 . . . . .	*	名前、*, *ALLAVL...
オブジェクト . . . . .	*NONE	
ディレクトリー・サブツリー . . . . .	*ALL	*ALL, *NONE
終り		
F3= 終了 F4= プロンプト F5= 最新表示 F12= 取り消し F13= この画面の使用法		
F24= キーの続き		

図 4-4 \*ALLUSR ライブラリー内のプログラム・オブジェクトを変換するために使用された STROBJCVN OPTION(\*CONVERT)

統合ファイル・システム内の Java プログラムは、オブジェクト (OBJ) パラメーターを使用して変換できます。オブジェクトのパス名は、単純名でも修飾名でも構いません。ワイルドカード文字を使用したパターンは、パス名の最後の部分で指定できます。/ を指定すると、ルート・ファイル・システムをアドレス指定することになります。

図 4-5 は、\*CONVERT オプションを使用した STROBJCVN コマンドの例を示しています。この例では、コマンドは、ルート・ファイル・システム内にある、変換が必要なすべての Java プログラムの変換を開始します。

オブジェクト変換開始 (STROBJCVN)		
選択項目を入力して、実行キーを押してください。		
オプション . . . . .	*CONVERT	*CONVERT, *CHECK
ライブラリー . . . . .	> *NONE	名前、*ALLUSR,*NONE
オブジェクト・タイプ . . . . .	*ALL	*ALL, *FILE, *ALLPGM, *SQL
ASP 装置 . . . . .	*	名前、*, *ALLAVL...
オブジェクト . . . . .	> @/@	
ディレクトリー・サブツリー . . . . .	*ALL	*ALL, *NONE
終り		
F3= 終了 F4= プロンプト F5= 最新表示 F12= 取り消し F13= この画面の使用法		
F24= キーの続き		

図 4-5 ルート・ファイル・システム内の Java プログラムを変換するための STROBJCVN OPTION(\*CONVERT)

STROBJCVN OPTION(\*CONVERT) は、この活動を要約した CPCB0E2 メッセージで完了するはずですが、図 4-6 は、このメッセージの例を示しています。

```
Additional Message Information

Message ID . . . . . : CPCB0E2      Severity . . . . . : 00
Message type . . . . . : Completion
Date sent . . . . . : 11/01/07     Time sent . . . . . : 22:27:44

Message . . . . . : Conversion of objects was successful.
Cause . . . . . : Start Object Conversion(STROBJCVN) OPTION(*CONVERT) ended with the
following results.

2 objects in library QAPTL have been processed. Some objects may not have been converted
because they were converted at an earlier time. They are included in this count. 0 objects
in *N have been processed. Some objects may not have been converted because they were
converted at an earlier time. They are included in this count. If *N or hex zero appears for
any fields, then this information does not apply.

                                                                    Bottom

Press Enter to continue.

F3=Exit  F6=Print  F9=Display message details
F10=Display messages in job log  F12=Cancel  F21=Select assistance level
```

図 4-6 STROBJCVN \*CONVERT 活動を要約したメッセージ CPCB0E2 の例

### 4.3 復元時のプログラム・オブジェクトの変換

復元時強制変換 (QFRCCVNRST) システム値を使用して、ユーザーは復元時に以下のオブジェクトを変換するかどうかを指定できます。

- ▶ プログラム (\*PGM)
- ▶ サービス・プログラム (\*SRVPGM)
- ▶ SQL パッケージ (\*SQLPKG)
- ▶ モジュール (\*MODULE)

QFRCCVNRST に指定できる値は、0 から 7 です。デフォルト値は 1 です。V6R1 互換形式への変換が必要なオブジェクトの変換を開始するには、2 の値で十分です。

このシステム値は、一部のオブジェクトの復元を防止することもできます。システム値によって変換するように指定されたが、十分な作成データを持っていないために変換できないオブジェクトは、復元されません。

**覚え書：** システム値 QFRCCVNRST を変更した場合、復元活動の後で、以前に使用していた値にリセットするのを忘れないでください。

復元コマンド (RST、RSTLIB、RSTOBJ、および RSTLICPGM) の強制オブジェクト変換 (FRCOBJCVN) パラメーターの \*SYSVAL 値は、QFRCCVNRST システム値で指定された値を使用します。図 4-7 は、ライブラリー復元 (RSTLIB) コマンドの強制オブジェクト変換パラメーターを示しています。

ライブラリー復元 (RSTLIB)		
選択項目を入力して、実行キーを押してください。		
保管された日付 . . . . .		日付
保管された時刻 . . . . .		時刻
オブジェクトの差異を許可 . . . . .	*NONE	*NONE, *ALL, *AUTL...
値の続きは +		
<b>強制オブジェクト変換:</b>		
復元時に変換 . . . . .	*SYSVAL	*SYSVAL, *NO, *YES
変換するオブジェクト . . . . .		*RQD, *ALL
ライブラリーに復元 . . . . .	*SAVLIB	名前、*SAVLIB
ASP 装置に復元 . . . . .	*SAVASPDEV	名前、*SAVASPDEV
ASP 番号に復元 . . . . .	*SAVASP	1-32, *SAVASP
出力 . . . . .	*NONE	*NONE, *PRINT, *OUTFILE
出力を受け取るファイル . . . . .		名前
ライブラリー . . . . .	*LIBL	名前、*LIBL, *CURLIB
<b>出力メンバー・オプション:</b>		
出力を受け取るメンバー . . . . .	*FIRST	名前、*FIRST
レコードを置換または追加 . . . . .	*REPLACE	*REPLACE, *ADD
続く . . .		
F3= 終了 F4= プロンプト F5= 最新表示 F12= 取り消し F13= この画面の使用法		
F24= キーの続き		

図 4-7 ライブラリー復元 (RSTLIB) コマンドの強制オブジェクト変換パラメーター

ユーザーは、システム値 QFRCCVNRST を変更することにより、システム全体の変換をオン/オフに切り替えることができます。ただし、復元コマンドの FRCOBJCVN パラメーターは、場合によってはシステム値を指定変更することがあります。例えば、次のような場合です。

- ▶ FRCOBJCVN パラメーターで \*YES と \*ALL を指定すると、QFRCCVNRST システム値のすべての設定を指定変更します。
- ▶ FRCOBJCVN パラメーターで \*YES と \*RQD を指定すると、このシステム値に対して 2 の値を指定したのと同じことになり、QFRCCVNRST システム値が 0 または 1 に設定されている場合は、これを指定変更します。

**注:**

- ▶ 一部のプロダクトのインストールに使用されるロード実行 (LODRUN) コマンドは、復元コマンドを使用しますが、QFRCCVNRST システム値を指定変更しません。変換を必要とし、かつインストールに LODRUN を使用するプロダクトは、システム値 QFRCCVNRST を 2 以上に設定することにより、インストール時の変換が可能になります。
- ▶ V6R1 から保管された、ターゲット・リリースが V5R4 または V5R3 のプログラム・モデル・オブジェクトは、受動システムでの復元時に V6R1 形式から変換されます。前提条件 PTF については、13 ページの 3.1.2、『必要な PTF』を参照してください。これらのプログラム・モデル・オブジェクトが V5R4 または V5R3 システムに復元された後は、それ以上のユーザー処置は必要ありません。



復元時の変換は、復元メディアとして使用されている装置に応じて、一度に3つか4つが同時に実行されることがあります。したがって、この変換方式は、最初の呼び出しでの変換方式に比べて、プロセッサ機能をより効果的に使用できる可能性があります。

## 4.4 最初のプログラムの実行時または呼び出し時の変換

復元時または STROBJCVN コマンドの実行時に変換されなかったプログラム・オブジェクトは、そのプログラムが最初に実行されたとき、または呼び出されたときに、自動的に変換されます。大きいプログラムの場合、このプロセスは実行に長時間かかることがあります。お客様が長い変換時間によって生じるパフォーマンス問題を懸念する場合は、実動に使用する前に、STROBJCVN コマンドを使用してプログラム・オブジェクトを変換することをお勧めします。詳しくは、47 0y°[0w° 4.2°A°wSTROBJCVN °i0g%p°N°6%îY†°x を参照してください。

**重要：**最初の呼び出しでの変換は、特に複数のプロセッサが利用可能な区画では、最も遅い変換方式になる場合があります。実際の結果は、呼び出しの順序や、まだ未変換のプログラムを同時に呼び出すジョブの数によって異なります。このことが、他の変換方式のいずれかを選択させる理由の1つになります。

## 4.5 変換時間とその改善方法

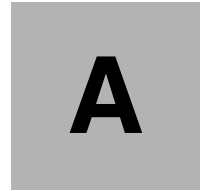
ANZOBJCVN コマンドで報告される変換時間は見積もりです。この見積もりは、システムのプロセッサ・フィーチャーに基づいています。実際の変換時間は見積もりとは異なる場合があります。変換時間には、次のようなさまざまな要因が影響を与える可能性があります。

- ▶ プロセッサの数
- ▶ 接続されているディスク・アーム
- ▶ 使用可能メモリー
- ▶ 変換ジョブの優先順位
- ▶ システム上の他のジョブ
- ▶ 並行変換ジョブの数

ほとんどの場合、オブジェクト変換は短時間で済み、容易に達成できると予想されます。しかし、場合によっては、変換の見積もり時間がユーザーの希望より長くなる場合があります。以下の提案に従うと、実際の変換時間を削減するのに役立つ可能性があります。

- ▶ \*CONVERT オプションを指定した STROBJCVN コマンドは、実行に長時間かかることがあります。すべてのシステムについて、特に対話式 (5250 OLTP) 機能が限定されているシステムでは、このコマンドをバッチ・ジョブとして実行することをお勧めします。
- ▶ 複数のプロセッサを備えた区画では、STROBJCVN が同時変換の機会を最も効果的に活用できる可能性が高い方式です。復元時の変換がその次に有効な方式で、最初の呼び出し時の変換は、複数のプロセッサを活用するには最も効果的でないと考えられます。
- ▶ ライブラリー内のオブジェクトの変換と、統合ファイル・システム内のオブジェクトの変換を、別々のジョブで実行することを検討してください。多数のライブラリー内の多数のオブジェクトを変換する必要がある場合、あるいは、変換が必要であるかどうかに関係なく、ライブラリー内に多数のファイル・オブジェクトが存在する場合は、ライブラリー内のオブジェクトを処理するために複数のジョブを実行することを検討してください。

- ▶ 多数のファイルまたは多数のメンバーを持つファイルを含むライブラリーが複数ある場合は、ライブラリーごとに別々のジョブを実行することを検討してください。ライブラリー内のファイルも一種の変換が行われる場合があり、これに多大な時間がかかることがあります。
- ▶ 変換が実行されている間、システム上にアクティブの他のユーザー・ジョブがない場合、最良のパフォーマンスを達成できます。
- ▶ STROBJCVN コマンドでは (\*ALLUSR 以外は) 1 つのライブラリーしか指定できないので、\*ALLUSR の一部ではない多数のライブラリー内のオブジェクトを変換する必要がある場合は、複数のバッチ・ジョブをサブミットする単一の CL プログラムを作成し、それぞれのバッチ・ジョブで 1 つのライブラリーを変換するようにすると便利な場合があります。
- ▶ 区画に分割された環境では、動的ロジカル・パーティショニングを利用して、使用可能なプロセッサ数および使用可能メモリー量を一時的に増やすことを検討してください。
- ▶ テストの結果、Lotus Domino と、特に QDOMINOxxx ライブラリー内のオブジェクトは、変換に長時間かかる可能性があることが実証されています。Domino の古いバージョンはすべてアップグレード前に除去し、QDOMINOxxx ライブラリー内のオブジェクトの変換は、別個のバッチ・ジョブで実行することを検討してください。



## MRI PTF

オブジェクト変換分析 (ANZOBJCVN) CL コマンドは、V5R3 用または V5R4 用の PTF として配布されています。この機能を使用するためには、MRI PTF を発注する必要があります。使用する言語用の MRI PTF を発注すると、必要なコード PTF も発注されます。ただし、場合によっては、コード PTF を別途発注することが必要な場合があります。必要な PTF に関する追加の考慮事項については、13 ページの 3.1.2、『必要な PTF』を参照してください。

表 A-1 は、i5/OS V5R3 用および V5R4 用のすべての MRI PTF をリストしています。

表 A-1 ANZOBJCVN コマンド用の MRI PTF

プロダクト・ロード	言語	V5R3	V5R4
2924	英語 (大 / 小文字)	SI28425	SI28415
2902	エストニア語	SI28472	SI28416
2903	リトアニア語	SI28473	SI28417
2904	ラトビア語	SI28474	SI28418
2905	ベトナム語	SI28475	SI28419
2906	ラオ語	SI28476	SI28420
2909	ベルギー英語	SI28477	SI28421
2911	スロベニア語	SI28478	SI28422
2912	クロアチア語	SI28479	SI28423
2913	マケドニア語	SI28480	SI28424
2914	セルビア語 (キリル文字)	SI28482	SI28426
2922	ポルトガル語	SI28483	SI28427
2923	オランダ語	SI28484	SI28428
2925	フィンランド語	SI28485	SI28429
2926	デンマーク語	SI28486	SI28431

プロダクト・ロード	言語	V5R3	V5R4
2928	フランス語	SI28488	SI28432
2929	ドイツ語	SI28489	SI28433
2930	日本語 (大 / 小文字)	利用不可	SI30602
2931	スペイン語	SI28490	SI28435
2932	イタリア語	SI28491	SI28436
2933	ノルウェー語	SI28492	SI28437
2937	スウェーデン語	SI28493	SI28438
2938	英語大文字 (DBCS (KANJI) 用)	SI28494	SI28439
2939	ドイツ語 MNCS	SI28495	SI28440
2940	フランス語 MNCS	SI28496	SI28441
2942	イタリア語 MNCS	SI28497	SI28442
2950	英語大文字	SI28498	利用不可
2954	アラビア語	SI28499	SI28443
2956	トルコ語	SI28500	SI28444
2957	ギリシャ語	SI28501	SI28445
2958	アイスランド語	SI28502	SI28446
2961	ヘブライ語	SI28503	SI28447
2962	日本語 (DBCS)	SI28504	SI28448
2963	オランダ語 MNCS	SI28505	SI28449
2966	ベルギー・フランス語	SI28506	SI28450
2972	タイ語	SI28512	SI28451
2974	ブルガリア語	SI28513	SI28452
2975	チェコ語	SI28514	SI28453
2976	ハンガリー語	SI28515	SI28454
2978	ポーランド語	SI28516	SI28455
2979	ロシア語	SI28517	SI28456
2980	ポルトガル語	SI28518	SI28457
2981	カナダ・フランス語 MNCS	SI28519	SI28458
2984	英語大 / 小文字 (DBCS 用)	SI28520	SI28459
2986	韓国語 (DBCS)	SI28521	SI28460
2987	中国語 (繁体字)	SI28522	SI28461
2989	中国語 (簡体字)	SI28523	SI28462
2992	ルーマニア語	SI28524	SI28463
2994	スロバキア語	SI28525	SI28464

プロダクト・ロード	言語	V5R3	V5R4
2995	アルバニア語	SI28527	SI28465
2996	ポルトガル語 MNCS	SI28528	SI28466
2998	ペルシア語	SI28529	SI28467



## ファイルのフィールド定義の一覧表

この付録では、オブジェクト変換分析 (ANZOBJCVN) コマンドによって生成されるファイルの各フィールドについて説明します。このフィールド情報を使用して QUERY を定義すると、提供されたデータをより良く理解するのに役立ちます。

ANZOBJCVN コマンドによって収集されるデータは、QUSRSYS ライブラリー内の 4 つの異なるファイルに保管されます。

- ▶ QSYS 内のファイル QAIZACVN は、マスター・テンプレート・ファイルで、収集されたデータは追加されません。ただし、QUSRSYS 内の QAIZACVN は、ライブラリー内の分析データ収集用のターゲット・ファイルです。
- ▶ ファイル QAIZADIR は、オブジェクト・パラメーターが使用された場合のディレクトリー情報のターゲットです。
- ▶ ファイル QAIZAOBJ は、オブジェクト・パラメーターが使用された場合の統合ファイル・システム・オブジェクト情報のターゲットです。
- ▶ ファイル QAIZASPL は、スプール・ファイルの収集情報のターゲットです。

**重要：**それぞれのファイルには、最後の収集からのデータのみが入っています。したがって、後続の収集は、直前にこれらのファイルに収集されたデータを置き換えることになります。後で収集を分析するために、これらのファイルを別のライブラリーにコピーしておくことができます。また、必要に応じて、これらのファイルに対して QUERY を実行し、追加の報告書を生成することもできます。QUERY の作成についての詳細は、35 ページの 3.5、『分析データに対する QUERY』を参照してください。

## QAIZACVN: 分析されたライブラリーの情報

表 B-1 は、QUSRSYS ライブラリー内の QAIZACVN ファイルのフィールド定義をリストしています。QAIZACVN ファイルは、ライブラリー内の分析データ収集用のターゲット・ファイルです。

表 B-1 QAIZACVN ファイルのフィールド定義

フィールド名	説明	属性
DIOBLI	ライブラリー名	Char(10)
DIOBNM	オブジェクトの名前	Char(10)
DIOBTP	オブジェクト・タイプ	Char(7)
DIOBAT	オブジェクト属性	Char(9)
DIOBSZ	オブジェクト・サイズ (バイト)	Packed (15,0)
DINOSZ	現在は未使用	Packed (15,0)
DIOBTX	テキスト記述	Char(50)
DISTIN	状況標識 : ▶ 0 = 正常 ▶ 1 = ロック ▶ 2 = 損傷	Char(1)
DIOBOW	オブジェクト所有者	Char(10)
DIOBAS	オブジェクト ASP	Char(5)
DIASPN	ASP 資源名	Char(10)
DISTGF	記憶域解放 : ▶ 0 = いいえ ▶ 1 = はい	Char(1)
DICRTD	プログラム作成データ域標識 : ▶ 0 = いいえ ▶ 1 = はい ▶ 2 = 適用外	Char(1)
DICOMP	オブジェクトを保管できる最古のリリース	Char(7)
DISFIL	ソース・ファイル	Char(10)
DISLIB	ソース・ライブラリー	Char(10)
DISMBR	ソース・メンバー	Char(10)
DISMBRE	ソース・メンバー存在 : ▶ 0 = いいえ ▶ 1 = はい	Char(1)
DIPPID	プログラム・プロダクト	Char(7)
DIREAS	理由コード	Char(7)
DILCEN	変更の世紀 : ▶ 0 = 19xx を示す ▶ 1 = 20xx を示す	Char(1)
DILDAT	変更日。形式は mmddyy。	Char(6)



フィールド名	説明	属性
DILTIM	変更時刻	Char(6)
DIUCEN	最終使用の世紀： ▶ 0 = 19xx を示す ▶ 1 = 20xx を示す	Char(1)
DIUDAT	最終使用日。形式は mmddyy。	Char(6)
DIUDTC	使用データの収集： ▶ 0 = いいえ ▶ 1 = はい	Char(1)
DIDUSD	使用回数 ( 日数 )	Packed(5,0)
DICUCR	使用回数リセットの世紀： ▶ 0 = 19xx を示す ▶ 1 = 20xx を示す	Char(1)
DIDUCR	使用回数リセット日。形式は mmddyy。	Char(6)
DIALCH	プログラムによる変更を許可： ▶ 0 = いいえ ▶ 1 = はい	Char(1)
DISCEN	保管の世紀： ▶ 0 = 19xx を示す ▶ 1 = 20xx を示す	Char(1)
DISDAT	保管日。形式は mmddyy。	Char(6)
DISTIM	保管時刻	Char(6)
DIACEN	保管アクティブの世紀： ▶ 0 = 19xx を示す ▶ 1 = 20xx を示す	Char(1)
DIADAT	保管アクティブ日。形式は mmddyy。	Char(6)
DIATIM	保管アクティブ時刻	Char(6)
DIRCEN	復元の世紀： ▶ 0 = 19xx を示す ▶ 1 = 20xx を示す	Char(1)
DIRDAT	復元日。形式は mmddyy。	Char(6)
DIRTIM	復元時刻	Char(6)
DISCMD	保管コマンド	Char(10)
DISAVF	保管ファイル	Char(10)
DISAVL	保管ファイル・ライブラリー	Char(10)
DIFLAB	ファイル・ラベル	Char(17)
DISDEV	保管装置： ▶ 0 = 保管ファイル ▶ 1 = テープ ▶ 2 = ディスケット	Char(1)
DISEQN	シーケンス番号	Zoned(8,0)

フィールド名	説明	属性
DIVOL1	ボリューム 1 を保管。オブジェクトを保管するために使用されたテープ、ディスク、または光ディスク・ボリューム。最大 10 個のボリュームを保管できる。	Char(6)
DIVOL2	ボリューム 2 を保管	Char(6)
DIVOL3	ボリューム 3 を保管	Char(6)
DIVOL4	ボリューム 4 を保管	Char(6)
DIVOL5	ボリューム 5 を保管	Char(6)
DIVOL6	ボリューム 6 を保管	Char(6)
DIVOL7	ボリューム 7 を保管	Char(6)
DIVOL8	ボリューム 8 を保管	Char(6)
DIVOL9	ボリューム 9 を保管	Char(6)
DIVL10	ボリューム 10 を保管	Char(6)
DIMORV	追加ボリューム標識： ▶ 0 = いいえ ▶ 1 = はい	Char(1)
DICEN	作成の世紀： ▶ 0 = 19xx を示す ▶ 1 = 20xx を示す	Char(1)
DICDAT	作成日。形式は mmddyy。	Char(6)
DICTIM	作成時刻	Char(6)
DICUSR	ユーザー作成	Char(10)
DICSYS	システム作成	Char(8)
DICPRS	圧縮状況： ▶ X = 不適格 ▶ N = いいえ ▶ F = 記憶域解放 ▶ T = 一時 ▶ Y = はい	Char(1)
DIUSDA	ユーザー定義属性	Char(10)
DIODMN	オブジェクト・ドメイン	Char(2)
DIOSTA	オブジェクト状態： ▶ X'8000' = *SYSTEM ▶ X'0001' = *USER ▶ X'0000' = *INHERIT	Char(2)
DIAUDV	オブジェクト監査値	Char(10)
DIOFSZ	オフライン・サイズ (バイト)	Hex(8)
DIOVRF	オーバーフロー・オブジェクト	Char(1)
DIFILT	ファイル・タイプ： S = ソース	Char(1)

フィールド名	説明	属性
DINMEM	メンバーの数	Binary(2)
DIOCTM	オブジェクト変換時間 ( ミリ秒 )	Packed(12,0)
DIOSGN	オブジェクトの署名状態： ▶ X'00' = 署名なし ▶ X'01' = 署名入り	Char(1)
DIOPRF	オブジェクト・プロファイル・データ： ▶ X'00' = プロファイル・データ使用不可 ▶ X'01' = プロファイル・データ使用可能	Char(1)
DIRESV	予約済み	Char(1)

## QAIZADIR: 分析されたディレクトリーの情報

表 B-2 は、QUSRSYS ライブラリー内の QAIZADIR ファイルのフィールド定義をリストしています。QAIZADIR ファイルは、ディレクトリー内の分析データ収集用のターゲット・ファイルです。

表 B-2 QAIZADIR ファイルのフィールド定義

フィールド名	説明	属性
QIZADIRIDX	パス名の ID ( ディレクトリーにのみ適用 )	Binary(4)
QIZADIRNAM1	親ディレクトリー・パス。パス長さが 1 KB を下回る場合にのみ使用。	VARGRAPHIC(1024)
QIZADIRNAM2	親ディレクトリー・パス。パス長さが 1 KB を上回る場合にのみ使用。最長 16 Mb の長さまで保管可能。	DBCLOB(16M)
QIZADRCCSID	ディレクトリーの CCSID	Binary(4)
QIZADREGION	ディレクトリーのパス領域 ID	GRAPHIC(2)
QIZADLANGID	ディレクトリーのパス言語 ID	GRAPHIC(3)
QIZADIRLEN	ディレクトリーのパス名の長さ ( バイト )	Binary(4)
QIZADIRFID	ディレクトリーのファイル ID	GRAPHIC(35)
QIZADFID	ディレクトリーの 4 バイト・ファイル ID	Binary(4)
QIZADIRFSID	ディレクトリーのファイル・システム ID	Binary(8)
QIZADIRGID	生成 ID	Binary(8)

## QAIZAOBJ: 分析された統合ファイル・システム・オブジェクトの情報

表 B-3 は、QUSRSYS ライブラリー内の QAIZAOBJ ファイルのフィールド定義をリストしています。QAIZAOBJ ファイルは、統合ファイル・システム・オブジェクト内の分析データ収集用のターゲット・ファイルです。

表 B-3 QAIZAOBJ ファイルのフィールド定義

フィールド名	説明	属性
QIZADIRIDX	親ディレクトリー索引	Binary(4)
QIZAOBJNAM	オブジェクトの名前	VARGRAPHIC(1024)
QIZAOBJLEN	QIZAOBJNAM フィールドの長さ (バイト)	Binary(4)
QIZACRTTIM	オブジェクトの作成日時。形式は YYMMDDHHMMSS。	TIMESTAMP(UTC)
QIZAACCTIM	オブジェクト・データの最終アクセス日時。形式は YYMMDDHHMMSS。	TIMESTAMP(UTC)
QIZACHGTIMA	オブジェクト属性の最終変更日時。形式は YYMMDDHHMMSS。	TIMESTAMP(UTC)
QIZACHGTIMD	オブジェクト・データの最終変更日時。形式は YYMMDDHHMMSS。	TIMESTAMP(UTC)
QIZATRGREL	このオブジェクトの作成ターゲット・バージョン	GRAPHIC(10)
QIZAOWN	オブジェクト所有者	GRAPHIC(10)
QIZAASP	オブジェクトの保管先 ASP	Binary(2)
QIZABLKSIZE	ブロック・サイズ (バイト)	Binary(4)
QIZANLNK	オブジェクトのリンク数	Binary(4)
QIZAFID	オブジェクトのファイル ID	GRAPHIC(35)
QIZAFIDS	オブジェクトの 4 バイト・ファイル ID	Binary(4)
QIZAGENID	オブジェクトの生成 ID	Binary(8)
QIZAFSID	ファイル・システム ID	Binary(8)
QIZADOM	オブジェクトのドメイン	GRAPHIC(10)
QIZASTA	オブジェクトの状態	GRAPHIC(10)
QIZAPRFDTA	オブジェクトのプロファイル作成データの存在 : ▶ 0 = 存在しない ▶ 1 = 存在	GRAPHIC(1)
QIZAOPTLVL	Java プログラムの最適化レベル。有効値 : ▶ 0 = このオブジェクトは Java プログラムではない ▶ ゼロ以外 = Java プログラムの最適化レベル値	GRAPHIC(10)
QIZAUDATE	使用情報、最終使用日。形式は YYMMDDHHMMSS。	TIMESTAMP(UTC)
QIZAUDCOUNT	使用情報、使用回数 (日数)。	Binary(4)

フィールド名	説明	属性
QIZAPRMLNK	<p>オブジェクトが複数の名前を持っている場合、このフィールドは最初に検出された名前に対してのみ設定されます。オブジェクトがすでに存在するかどうかを知る方法は、そのオブジェクトのファイル・システム ID、ファイル ID、および生成 ID を検査することです。これらを一緒に使用すると固有になります。</p> <p>▶ 1 = 最初に検出された名前</p>	Binary(2)
QIZASIG	<p>オブジェクトが i5/OS デジタル署名を持っているかどうか。以下の値が有効です。</p> <p>▶ 0 = オブジェクトは i5/OS デジタル署名を持っていない</p> <p>▶ 1 = オブジェクトは i5/OS デジタル署名を持っている</p>	Binary(2)
QIZASYSSIG	<p>オブジェクトはシステムに信頼されたソースによって署名されたかどうか。以下の値が有効です。</p> <p>▶ 0 = システムに信頼されたソースからの署名はない</p> <p>▶ 1 = オブジェクトはシステムに信頼されたソースによって署名された。オブジェクトに複数の署名がある場合、少なくとも 1 つの署名は、システムに信頼されたソースからのものです。</p>	Binary(2)
QIZAMLSIG	<p>オブジェクトが複数の i5/OS デジタル署名を持っているかどうか。以下の値が有効です。</p> <p>▶ 0 = オブジェクトは 1 つだけデジタル署名を持っている</p> <p>▶ 1 = オブジェクトは複数のデジタル署名を持っている。QIZASYSSIG フィールドの値が 1 の場合、少なくとも 1 つの署名はシステムに信頼されたソースからのものです。</p>	Binary(2)
QIZADIRTY2	<p>ディレクトリー・オブジェクト (*DIR) の形式。以下の値が有効です。</p> <p>▶ 0 = ディレクトリー形式は *TYPE1。</p> <p>▶ 1 = ディレクトリー形式は *TYPE2。</p>	Binary(2)
QIZAFILTY2	<p>ストリーム・ファイル (*STMF) の形式。以下の値が有効です。</p> <p>▶ 0 = ストリーム・ファイル形式は *TYPE1。</p> <p>▶ 1 = ストリーム・ファイル形式は *TYPE2。</p>	Binary(2)
QIZACASE	<p>このオブジェクトを含むファイル・システムの大 / 小文字の区別を示します。以下の値が有効です。</p> <p>▶ 0 = ファイル・システムは大 / 小文字の区別をしない</p> <p>▶ 1 = ファイル・システムは大 / 小文字の区別をする</p>	Binary(2)
QIZACVNTIME	<p>オブジェクトの見積もり Java プログラム変換時間 (ミリ秒)。</p>	Packed(12,0)

フィールド名	説明	属性
QIZANAMFLG	<p>オブジェクト名には、ターゲット・リリースの新規の Unicode 大 / 小文字規則によって影響を受ける文字が含まれているかどうかを示します。以下の値が有効です。</p> <ul style="list-style-type: none"> <li>▶ 0 = オブジェクト名は影響を受ける文字を含んでいない</li> <li>▶ 1 = オブジェクト名は影響を受ける文字を含んでいる</li> </ul>	Binary(2)
QIZACVNFLG	<p>Java プログラム・オブジェクトは変換が必要かどうか。以下の値が有効です。</p> <ul style="list-style-type: none"> <li>▶ 0 = オブジェクトは変換が不要であるか、または Java プログラムではない。</li> <li>▶ 1 = オブジェクトは、変換を必要とする接続プログラムを含む .jar ファイル、.zip ファイル、または .class ファイルである。</li> </ul>	Binary(2)

## QAIZASPL: 分析されたスプール・ファイルの情報

表 B-4 は、QUSRSYS ライブラリー内の QAIZASPL ファイルのフィールド定義をリストしています。QAIZASPL ファイルは、分析されたスプール・ファイルの収集用のターゲット・ファイルです。

表 B-4 QAIZASPL ファイルのフィールド定義

フィールド名	説明	属性
SPDTTM	日付 (YYMMDD) と時刻 (HHMMSS)。収集されたデータの日時。	Char(12)
SPDCEN	世紀桁 (年号の上 2 桁): <ul style="list-style-type: none"> <li>▶ 0 = 19xx を示す</li> <li>▶ 1 = 20xx を示す</li> </ul>	Char(1)
SPSNAM	システム名	Char(8)
SPVRM	バージョン・リリース・モディフィケーション。 VxRyMz 形式でリリース・レベルを指定します。ここで、 Vx = バージョン番号 Ry = リリース番号 Mz = 修正レベル	Char(6)
SPASP	ASP 番号。ASP 番号を指定します。 <ul style="list-style-type: none"> <li>▶ 1 = システム ASP</li> <li>▶ 2 から 32 = 基本 ASP</li> <li>▶ 33 から 255 = 独立 ASP</li> </ul>	Packed(5,0)
SPASPN	ASP 装置名。ASP の名前を指定します。ブランクの値は、システム ASP または基本 ASP を指定します。	Char(10)
SPSPLN	スプール・ファイル。ASP で変換する必要がある、QSPLxxxx ライブラリー内のスプール・ファイルの数を指定します。	Packed(11,0)
SPTMCV	スプール・ファイルの変換時間。ASP 内のすべてのスプール・ファイルを変換するために必要な合計時間を秒数で指定します。	Packed(11,0)



## 変換を必要とする IBM ソフトウェア ・ スキップ・ シップ・ プロダクト

この付録では、V6R1 用の新規バージョンがなく、システムを V6R1 にアップグレードした後で変換する必要がある、IBM スキップ・ シップ・ ライセンス・ プログラム・ プロダクト (LPP) に関する情報を提供します。表 C-1 にリストされている LPP は、V6R1 上でサポートされるが、V6R1 用の新規バージョンがないために変換する必要があるスキップ・ シップ・ プロダクトです。

**重要:** V6R1 に移行する前に、そのリリースでサポートされない LPP をすべて除去しておくことをお勧めします。場合によっては、V6R1 にアップグレードするまで待っていると、これらのプロダクトを完全に除去するのが容易でなくなる場合があります。「ライセンス・ プログラムの処理 (Go LICPGM)」メニューには、システム上にインストールされたすべてのプログラム・ プロダクトが表示されるわけではありません。ソフトウェア資源表示 (DSPSFWRSC) コマンドを使用して、システムにインストールされているすべてのソフトウェア・ リソースをリストしてください。

LPP のプログラム変換は、プロダクト・ ライブラリー上で実行する必要があります。プロダクトを初めて使用する前に、ライブラリー上で変換を実行します。

これらのスキップ・ シップ・ プロダクトのいずれかを V6R1 にインストールする場合は、インストール時にプロダクト・ ライブラリー内のオブジェクトを変換することをお勧めします。これは、RSTLICPGM コマンドで強制オブジェクト変換 (FRCOBJCVN) パラメーターを使用することによって達成できます。詳細については、55 ページの 4.3、『復元時のプログラム・ オブジェクトの変換』を参照してください。

スキップ・ シップ・ プロダクトがアップグレード前にインストールされた場合、プロダクト・ ライブラリー内のオブジェクトは、i5/OS アップグレード・ プロセスの一部として自動的に変換されません。この場合は、STROBJCVN コマンドを使用して変換を行うことをお勧めします。これらのプロダクトの変換を計画する際には、いくつかの要因を考慮する必要があります。オブジェクト変換について詳しくは、45 ページの第 4 章、『プログラム変換方式』を参照してください。

変換プロセス中にメッセージ CPIB0DC “Digital signature lost for object” (オブジェクトのデジタル署名が失われました) を受け取った場合は、デジタル署名が除去されました。デジタ

ル署名は通常、プロダクトの実行前にのみ使用されるので、プロダクト機能には影響はありません。

**Domino 情報** : Domino プロダクト・ライブラリーには、多数のプログラムおよびサービス・プログラムが含まれており、その一部は非常に大きなものです。Domino の変換プロセスは、多大な時間がかかることがあります。Domino 固有の情報については、以下のアドレスにある Lotus Domino for i5/OS Support Web サイトを参照してください。

<http://www.ibm.com/servers/eserver/series/domino/support/>

表 C-1 V6R1 でサポートされるスキップ・シップ・プロダクト

プロダクト ID	バージョン	プロダクト・ライブラリー	プロダクト名
5722BZ100	V5R1M0	QIBS	IBM Business Solutions
5722IA100	V5R2M0	QIA1BASE	IBM Integration Assistant for iSeries
	V5R2M0	QIA1OPT13	Business Integration Connect – Express 4.2
	V5R2M0	QIA1OPT16	WebSphereR Partner Gateway – Express
5722IP100	V5R3M0	QIPS	IBM Infoprint Server for iSeries
5722XP100	V5R3M0	QIWR/IFS	System i Access for Wireless
5723J2500	V7M0	QDOMDOC	Lotus Domino Document Manger
5724H7200	V6R0M0	QMQM QMQMSAMP	WebSphere MQSeriesR for iSeries MQ, V6
5724J2300	V7R5M1	QSAMETIME	Lotus Sametime Instant Messaging
5724J2400	V7R0M0	QQPLACE	Lotus QuickPlace
5724S31	V8.0.0.2	QQPLACE	Lotus Quickr™
5733FXD00	V4R5M0	QDOMFXD	Domino Fax
5733ID100	V4R5M0	QAFPDNR	Infoprint Designer
5733LD700	V7R0M3	QDOMINO703	Lotus Domino for iSeries V7.0.3
5733LD800	V8R0M1	QDOMINO801	Lotus Domino for iSeries V8
5733LEI	V6R5M4	QDOMINO703 または QDOMNO801	Lotus Enterprise Integrator®
5733W6000	V6R0M0	QWAS60	WebSphere Application Server for OS/400 V6
5733W6100	V6R1M0	QWAS61	WebSphere Application Server V6.1 for i5/OS
5733XT200	V1R1M0	QXMMLIB	XML Toolkit for IBM System i5™
5798FAX00	V5R2M0	QFAX	Facsimile Support
5799PTL0	V5R4M0	QAPTL	iSeries Tools for Developers



# 関連資料

ここでは、この Redpaper で取り上げたトピックについて、さらに詳しく説明する資料を紹介いたします。

## その他の資料

以下の資料も関連のある追加情報を提供します。

- ▶ *i5/OS および関連ソフトウェアのインストール、アップグレードおよび削除* (SD88-5002-10)  
<http://publib.boulder.ibm.com/infocenter/systems/topic/rzahc/sc415120.pdf>
- ▶ Reclaim Spooled File Storage Experience Report  
<http://publib.boulder.ibm.com/infocenter/iseriess/v5r4/topic/experience/spoolreclaim.pdf>
- ▶ *System i ILE 概念バージョン 6 リリース 1* (SD88-5033)  
<http://publib.boulder.ibm.com/infocenter/systems/topic/books/sc415606.pdf>

## オンライン・リソース

以下の Web サイトも追加の情報源としてご利用いただけます。

- ▶ Informational APAR II14306 on Analyze Object Conversion  
[http://www-912.ibm.com/n\\_dir/nas4apar.nsf/c79815e083182fec862564c00079d117/3af47a966c4df94586257306003c6868?OpenDocument&Highlight=2,ii14306](http://www-912.ibm.com/n_dir/nas4apar.nsf/c79815e083182fec862564c00079d117/3af47a966c4df94586257306003c6868?OpenDocument&Highlight=2,ii14306)
- ▶ System i i5/OS プログラム資料説明書 バージョン 6 リリース 1  
<http://publib.boulder.ibm.com/infocenter/systems/scope/i5os/topic/rzaq9/rzaq9.pdf>
- ▶ System i Upgrade planning  
<http://www-304.ibm.com/systems/support/i/planning/upgrade/index.html>

## Redbook の入手方法

Redbook、Redpaper、技術情報、ドラフト資料、および補足資料は、次の Web サイトで検索、表示、またはダウンロードすることができます。また、Redbook のハードコピーを発注することもできます。

[ibm.com/redbooks](http://ibm.com/redbooks)

## IBM のサポート

IBM Support and downloads

[ibm.com/support](http://ibm.com/support)

IBM Global Services

[ibm.com/services](http://ibm.com/services)





# IBM i5/OS Program Conversion 使用の手引き (i5/OS V6R1 用)



Redpaper™

以前のリリースで  
の i5/OS V6R1 プロ  
グラム変換の準備

オブジェクト変換  
分析  
(ANZOBJCVN)  
ツールの使用

i5/OS V6R1 で実行  
するためのプログ  
ラムの変換

IBM は、2008 年 3 月に i5/OS V6R1 (IBM i 6.1) の提供を開始しました。V6R1 では、IBM はマシン・インターフェース (MI) アーキテクチャーの機能を活用して、プログラムを大幅に改善しています。プログラムは、パフォーマンスの向上、オペレーティング・システムおよびプロセッサの一連の新機能、および一層強化されたシステム保安全性から利益を得ることができます。これらの機能強化を使用可能にするには、以前のリリース用に作成されたすべての MI プログラムを V6R1 で実行できるように変換する必要があります。MI プログラムには、統合言語環境 (ILE) プログラムとオリジナル・プログラム・モデル (OPM) プログラムが含まれます。

プログラムを変換するには、その作成データが使用可能でなければなりません。MI プログラムはデフォルトで作成データを保存するため、ほとんどのプログラムは変換が可能です。外部アクセスから作成データが除去された場合でも、プログラムとその構成モジュールのターゲット・リリースが V5R1 以降であれば、ライセンス内部コード (LIC) は作成データにアクセスできます。

V5R4 または V5R3 上でオブジェクト変換分析 (ANZOBJCVN) コマンドを実行すると、i5/OS アップグレードの計画を立てるのに役立ちます。ANZOBJCVN コマンドは、V6R1 で実行するように変換されないプログラムを識別し (ある場合)、プログラム変換に要する時間を見積もります。また、他の 2 つのタイプの変換 (統合ファイル・システムの名前とスプール・ファイル) に関する情報も提供します。ANZOBJCVN コマンドへのアクセスは、本書に記載されている該当の PTF をロードして適用することにより可能になります。

この IBM Redpaper 資料は、システム管理者および現場技術者が V6R1 へのアップグレードまたはマイグレーションについて理解し、準備することを支援することを目的としています。本書では、プログラム変換とは何か、なぜ V6R1 上でプログラムを作成または再作成するのかについて説明します。また、V5R3 または V5R4 上で ANZOBJCVN ツールを使用して V6R1 オブジェクト変換の準備をする手順についても詳しく説明します。最後に、V6R1 で実行するようにプログラムを変換するためのプログラム変換方式について説明します。

INTERNATIONAL  
TECHNICAL  
SUPPORT  
ORGANIZATION

BUILDING  
TECHNICAL  
INFORMATION  
BASED ON  
PRACTICAL  
EXPERIENCE

IBM Redbooks は、IBM International Technical Support Organization で作成されています。世界各地からの IBM の専門家、お客様、およびパートナーが、実際のシナリオに基づいて、タイムリーなテクニカル情報を作成しています。お客様がご自身の環境で効率的に IT ソリューションを実装できるように、具体的な推奨事項を提示しています。  
追加情報：  
[ibm.com/redbooks](http://ibm.com/redbooks)

SG88-4039-00

(英文原典 : REDP-4293-00)