



Treinamento: Infra-Estrutura Protheus
Módulo 03: Análise de Eventos de Erro

Tópicos Abordados

1.	Objetivos	3
2.	Arquivos de Eventos de Erro	3
3.	Estrutura dos Eventos de Erro	4
3.1.	Exemplo da Estrutura do Erro	5
4.	Tipos de Eventos de Erro	7
4.1.	Definição das Convenções de Texto Utilizadas	7
4.2.	Definição dos tipos de erros	8
4.3.	Tipos de erros específicos	10
5.	Processos de Análise	12
5.1.	Análise Simples	12
5.2.	Análise com Consulta	12
5.3.	Análise Complexa	13
5.4.	Análise de Erros Específicos	13
5.4.1.	Análise de Erros na execução de Queries	13
5.4.2.	Análise de Erros na execução de Lançamentos Padrões	14
6.	Estudo de Casos	16
6.1.	Análise Simples	16
6.1.1.	Invalid Field Name	16
6.1.2.	Incorrect Sintaxe Near (query)	18
6.2.	Análise com Consulta	20
6.2.1.	Argument error	20
6.2.2.	Index not found	22
6.2.3.	Invalid Data Type	24
6.2.4.	Type mismatch on <operation>	25
6.2.5.	Variable does not exist	26
6.2.6.	Variable is not <tipo>	28
6.3.	Dicas para Análise de Erros Complexos	29
6.3.1.	Array out of bounds	29
6.3.2.	File is in EOF	30
6.3.3.	There is no current Index	31
7.	Casos de Análise	32
7.1.	Caso de Análise Simples	32
7.2.	Caso de Análise com Consulta	34
7.3.	Caso de Análise Complexa	37
7.4.	Caso de Análise Específica: Lançamentos Padrões	41
8.	Informações Complementares	43
8.1.	Variáveis Publicas do Sistema	43
8.2.	Estrutura do Ambiente Protheus	44
8.2.1.	Componentes e Dependências	44
8.2.2.	Formas de Atualização dos Componentes	45
9.	Referências	46
9.1.	Material Utilizado	46
9.2.	Colaboradores	46

1. Objetivos

Os objetivos deste documento são:

- ✓ Definir a estrutura dos arquivos error.log gerados pela aplicação Protheus;
- ✓ Definir, conceituar e diferenciar os tipos de erros descritos nestes arquivos;
- ✓ Definir um processo de análise simples para o entendimento dos erros;
- ✓ Exemplificar com casos reais os processos de análise.

Este documento não tem a pretensão de ser uma referência incontestável para a análise de eventos de erro, dada as inúmeras possibilidades de combinações que podem tornar um dos processos de análise aqui descritos ineficaz.

2. Arquivos de Eventos de Erro

Os arquivos error.log são gerados quando ocorrem não conformidades no sistema para as quais não existem tratamentos para situações não previstas nas rotinas.

Por esta razão, estes arquivos contêm a exata situação do sistema no momento do erro, possibilitando ao analista de sistemas ou de desenvolvimento detectar e solucionar a causa do mesmo, seja esta causa em um ambiente específico, como por exemplo, a base de dados de um cliente ou na própria aplicação.

Deve-se entender como "situação do sistema" os seguintes componentes:

- ✓ O conjunto de variáveis ativas e seus respectivos escopos (Locais, Private e Public) e conteúdos;
- ✓ As funções e rotinas principais, executadas até o momento do erro;
- ✓ Os parâmetros recebidos por cada função e suas respectivas variáveis (Privates e Locais);
- ✓ As tabelas e ALIAS ativos, especificando o índice, campos e conteúdo dos campos, e qual o ALIAS selecionado no momento.

Para complementar estas informações, na "pilha" de chamada de funções são especificadas cada uma das linhas de chamadas das funções e as datas dos respectivos fontes envolvidos.

3. Estrutura dos Eventos de Erro

1- Cabeçalho - Descrição do erro:

Contém a mensagem de erro, detalhando seu tipo e a linha no fonte que o ocasionou.

2- Pilha de Funções

Contém todas as funções principais executadas a partir da entrada do módulo até a geração do erro.

3- Variáveis Public

Lista todas as variáveis de escopo Public, seus tipos e respectivos conteúdos.

4- Detalhamento das funções

Contém, respeitando a ordem apresentada anteriormente na Pilha de Funções, o detalhamento da chamada de cada função, contendo:

- ✓ Parâmetros recebidos pela função;
- ✓ Tipo e Conteúdo das Variáveis Private;
- ✓ Tipo e Conteúdo das Variáveis Locais;

5- Tabelas

Contém a listagem de todas as tabelas ativas no módulo no momento do erro, apontando com ** o ALIAS ativo, com o seguinte detalhamento:

- ✓ ALIAS da Tabela;
- ✓ Nome físico do arquivo;
- ✓ Filtro aplicado a tabela;
- ✓ Total de Registros da tabela;
- ✓ RECNO do registro atualmente posicionado;
- ✓ Listagem dos Índices existentes para a tabela, apontado com ** o índice ativo;
- ✓ Listagem dos campos em uso na tabela, seus tipos e respectivos conteúdos.

3.1. Exemplo da Estrutura do Erro

1	AP6 Build 7.00.031210A ERRO THREAD (Administrator, CONDUMAX04) 14/01/2004 17:19:19 type mismatch on compare on VALORCTB(CTBXFUN.PRX) 27/11/2003 line : 4419
2	<pre> [build:7.00.031210A] [thread 888] Called from CTR180IMP(CTBR180.PRW) 28/10/2003 line : 789 Called from { LEND CTR180IMP(@LEND,WNREL,CSTRING,ASETOFBOOK,ACTBMOEDA,CSAYCC,NDIVIDE)}(CTB R180.PRW) line : 203 Called from line : 1519 Called from { EVAL(BACTION, @LSTOP),LSTOP := .T., ODLG:END() }(APLIB080.PRW) line : 1519 Called from line : 0 Called from RPTSTATUS(APLIB080.PRW) 04/12/2003 line : 1520 Called from CTBR180(CTBR180.PRW) 28/10/2003 line : 203 Called from __EXECUTE(APLIB090.PRW) 04/12/2003 line : 971 Called from { __EXECUTE('CTBR180()','XXXXXXXXXX','BALANC. CC X CTA ','')} line : 2752 Called from line : 0 Called from SIGAADV(SIGAADV.PRW) 23/04/2002 line : 223 </pre>
3	Variáveis em uso Publicas Publica 1: OMAINWND(O) :O Publica 2: LLEFT(L) :.F. Publica 3: CUSERNAME(C) :000344 MARCEL Publica 4: LQUERY(L) :.F. Publica 5: __MVCOMBOBOX(L) :.T.
4	<pre> SIGAADV(SIGAADV.PRW) 23/04/2002 Param 1: CEMPTRY(U) : NIL Param 2: CUSER(U) : NIL Param 3: LFULL(U) : NIL Private 1: OSHORTLIST(U) :NIL Private 2: NCAMPOS(N) :200 Local 1: CEMPTRY(U) :NIL Local 2: CUSER(U) :NIL Local 3: LFULL(U) :NIL Local 4: OICO(U) :NIL CTBR180(CTBR180.PRW) 28/10/2003 Private 1: ARETURN(A) : Private 2: ALINHA(A) : Private 3: CPERG(C) :CTR180 Private 4: NLASTKEY(N) :0 Private 5: NOMEPROG(C) :CTBR180 Local 1: ASETOFBOOK(A) : Local 2: ACTBMOEDA(A) : </pre>

5

Files

p:\ap_data\adv\oficial\sigamat.emp - Alias:SM0; Filter : ; Recno:3;
Total Recs:5; Order:1

Índice (0) : **SIGAMAT M0_CODIGO+M0_CODFIL

Campo 1: M0_CODIGO(C) :01

Campo 2: M0_CODFIL(C) :02

Campo 3: M0_FILIAL(C) :CONDUMAX I I

Campo 4: M0_NOME(C) :CIAFUNDI

Campo 5: M0_NOMECON(C) :ELETRO METALURGICA CIAFUNDI LTDA

Campo 6: M0_ENDCOB(C) :RD WILQUEM MANOEL NEVES KM 3 5

p:\ap_data\adv\oficial\sx3010.dbf - Alias: SX3; Filter : ;
Recno:8132; Total Recs:23363; Order:2

Índice (0) : SX30101 X3_ARQUIVO+X3_ORDEM

Índice (1) : **SX30102 X3_CAMPO

Índice (2) : SX30103 X3_GRPSXG+X3_ARQUIVO+X3_ORDEM

Índice (3) : SX30104 X3_ARQUIVO+X3_FOLDER+X3_ORDEM

Campo 1: X3_ARQUIVO(C) :SA1

Campo 2: X3_ORDEM(C) :26

Campo 3: X3_CAMPO(C) :A1_CGC

Campo 4: X3_TIPO(C) :C

Campo 5: X3_TAMANHO(N) :14

Campo 6: X3_DECIMAL(N) :0

Campo 7: X3_TITULO(C) :CGC/CPF

Campo 8: X3_TITSPA(C) :CNPJ/CPF

Campo 9: X3_TITENG(C) :CNPJ/CPF

Campo 10: X3_DESCRIC(C) :CGC/CPF do cliente

Campo 13: X3_PICTURE(C) :@R 99.999.999/9999-99

****CT1010 - Alias:CT1; Filter : ; Recno:1759; Total Recs:1991;
Order:1**

Índice (0) : **CT10101 CT1_FILIAL+CT1_CONTA

Índice (1) : CT10102 CT1_FILIAL+CT1_RES

Índice (2) : CT10103 CT1_FILIAL+CT1_CLASSE+CT1_CONTA

Índice (3) : CT10104 CT1_FILIAL+CT1_GRUPO+CT1_CONTA

Índice (4) : CT10105 CT1_FILIAL+CT1_CTASUP

Índice (5) : CT10106 CT1_FILIAL+CT1_DESC01

Índice (6) : CT10107 CT1_FILIAL+CT1_CC

Campo 1: CT1_FILIAL(C) :

Campo 2: CT1_CONTA(C) :3

Campo 3: CT1_DESC01(C) :D E S P E S A S

Campo 4: CT1_DESC02(C) :

Campo 5: CT1_DESC03(C) :

Campo 6: CT1_DESC04(C) :

Campo 7: CT1_DESC05(C) :

Campo 8: CT1_CLASSE(C) :1

Campo 9: CT1_NORMAL(C) :1

Campo 10: CT1_RES(C) :

Campo 11: CT1_BLOQ(C) :2

Campo 12: CT1_DTBLIN(D) :0/0/0

4. Tipos de Eventos de Erro

Neste tópico serão definidos os tipos de erros mais comuns e exemplificadas as situações aonde os mesmos normalmente ocorrem.

4.1. Definição das Convenções de Texto Utilizadas

Neste documento são utilizadas convenções de texto de forma a generalizar determinados tipos de informações. Estas informações são relacionadas a conteúdos que podem variar em diferentes situações mas que permanecem relacionados ao mesmo tipo de informação. Os tipos que se enquadram nestas convenções são identificados pelos caracteres "<>" conforme a listagem abaixo:

- <alias>
- <atributo>
- <caminho do arquivo + nome do arquivo>
- <campos da chave do índice>
- <conteúdo>
- <data do fonte>
- <expressão de filtro >
- <fonte>
- <função>
- <linha>
- <nome do campo>
- <nome_da_coluna>
- <número do campo>
- <número do índice>
- <número do registro posicionado>
- <operação>
- <operador>
- <ordem de índice ativa>
- <select_parcial>
- <tabela>
- <texto>
- <tipo do campo>
- <tipo>
- <total de registros na tabela>
- <variável>

4.2. Definição dos tipos de erros

Alias already in use

A rotina está definindo um novo alias, normalmente através da função DbUseArea, o qual já está em uso no sistema.

Este erro normalmente ocorre na criação de arquivos temporários, em rotinas de relatórios e manipulação de arquivos / browses e em rotinas que utilizam queries.

Argument error

As funções internas do sistema avaliam o tipo das variáveis utilizadas como parâmetros. Quando o tipo da variável passada como parâmetro para uma dessas funções não condiz com o tipo pré-definido na função ocorre o erro.

Este tipo de erro pode ocorrer nas mais diversas situações, mas está normalmente vinculado à:

- ✓ Tipo incorreto do campo no SX3;
- ✓ Tipo ou conteúdo incorreto do parâmetro "MV_" no SX6 ;
- ✓ Retorno incorreto de um Ponto de Entrada para uma variável em uso pela rotina padrão.

Array out of bounds

A posição solicitada de uma variável do tipo array não existe.

Este tipo de erro é genérico e normalmente de difícil análise, pois o conteúdo das variáveis do tipo array não é exibido no error.log, tornando necessária uma análise apurada do fonte para determinar se houve falha na criação, atribuição ou na utilização do mesmo.

File is in EOF

A rotina está efetuando alguma operação de leitura ou gravação em uma tabela a qual está em fim de arquivo.

Este é um dos erros mais genéricos, pois está normalmente vinculado a rotinas padrões customizadas através de pontos de entrada, os quais não restauram o ambiente do sistema após sua execução.

Incorrect Sintaxe near (Query)

A rotina está executando uma query através da função TcGenQuery(), a qual está com um erro na sua estrutura.

Este erro está associado à montagem incorreta dos argumentos da query pela rotina ou a conteúdos indevidos nos campos das tabelas associadas a query.

Importante: O símbolo (') – "aspas simples" é interpretado pelo banco de dados como um limitador de campos do tipo caractere. Um campo de uma tabela do sistema não pode ser preenchido, independente da posição, com este símbolo.

Exemplo: Campo SE1- >E1_PREFIXO: A1'

Index not Found

A rotina está selecionando através da função DbSetOrder() um índice que não existe definido no SINDEK / SIXxxx para o Alias corrente.

Uma situação comum em rotinas que criam índices temporários para manipulação dos arquivos, ou na utilização do cadastro de Relacionamentos do SigaCTB, pois nos casos em que o número do índice é superior à 9, o campo na tabela "CTL" deve ser alterado para duas posições e não ser tratado como "A", "B" ou afins.

Invalid Cursor State (MSSQL e INFORMIX)

Apesar da mensagem de erro ser exibida no sistema Microsiga, o erro está relacionado à falta de recursos no banco, sendo que no manual da ferramenta ou no help está descrito o procedimento para estas situações.

Invalid Data Type

Este erro ocorre na manipulação de campos de tabelas que estão com um tipo de dado diferente do esperado pelo sistema.

Normalmente vinculado a ambientes de bancos de dados, onde campos data não estão definidos da tabela de controle do sistema chamada TOP_FIELD.

Invalid Field Name

A rotina está utilizando um campo que não existe na tabela em uso.

Esta situação normalmente ocorre após processos de atualização de versão ou em ambientes novos, os quais foram criados com base em um SXSBR.txt desatualizado ou incorreto.

There is no current Index

A rotina está selecionando através da função DbSetOrder() um índice sendo que o Alias corrente não possui nenhum índice ativo.

Esta situação é comum em rotinas que manipulam incorretamente o ALIAS, eliminando os índices ativos.

Type mismatch on <operação>

A rotina está efetuando uma operação algébrica ou uma comparação nas quais os tipos de variáveis não são compatíveis.

Variable does not exist

A rotina está utilizando uma variável que não foi pré-definida.

Aparentemente uma situação de erro simples, mas que pode ser causada por:

- ✓ Especificação incorreta do escopo da variável no desenvolvimento da rotina;
- ✓ A rotina foi executada de uma forma não tradicional, o que provocou a não inicialização da variável;
- ✓ Em lançamentos contábeis, o alias não foi especificado na definição de campos ou não foram respeitadas as diferenças para aquela contabilização entre o modo On-Line e Off-Line.

Variable is not <tipo>

A rotina está utilizando uma propriedade específica de uma variável ou um objeto, só que o tipo da variável não condiz com a propriedade utilizada.

Esta situação ocorre devido a inicialização incorreta da variável, com um tipo que não condizia com o esperado pelo programa. As causas mais comuns deste tipo de erro são retornos incorretos de Pontos de Entrada e a definição incorreta de parâmetros "MV_" na tabela SX6.

4.3. Tipos de erros específicos

Os erros específicos tratados neste documento são referentes a duas situações que, apesar do problema ser sempre focado em um mesmo ponto, as mensagens de erro exibidas pelo Protheus variam em quantidade e formas de interpretação.

Estes dois tipos de erros são:

- ✓ Erros na execução de Queries;
- ✓ Erros na execução de Lançamentos Padrões – Funções DETPROVA e CTRELATION.

Erros na execução de queries

Apesar da grande diversidade de mensagens, todas apontam sempre para problemas na sintaxe do comando submetido ao banco pelo TopConnect. Desta forma em todas as situações deverá ser identificada e analisada a Query.

Alguns dos tipos de erros mais comuns que se enquadram nesta situação estão listados a seguir:

- ☐ Attribute **<atributo>** not found;
- ☐ Column **<nome_da_coluna>** not in specified tables;
- ☐ **<comando_sql>** keyword not found where expected;
- ☐ Incorrect syntax near **<texto>**;
- ☐ Invalid character;
- ☐ Invalid column name **<nome_da_coluna>**;

- ☐ Query Error on **<função>** (**<fonte>**) **<data do fonte>** line : **<linha>**
- ☐ The data type of an operand of an arithmetic function or operation "+" is not numeric;
- ☐ The data types of the operands for the operation **<operador>** are not compatible;

Erros na execução de Lançamentos Padrões – Função DETPROVA

A maior parte dos erros referentes à função DETPROVA estão relacionados com problemas na configuração dos cadastros dos Lançamentos Padrões.

Isto ocorre porque os conteúdos dos campos das tabelas SI5 e CT5, com algumas restrições, são interpretados por esta função. Desta forma, um conteúdo inválido irá causar um erro fatal de execução.

Os tipos de erros que normalmente ocorrem por problemas nas configurações dos Lançamentos Padrões são:

- ☐ Argument error
- ☐ File is in EOF
- ☐ Index not found
- ☐ Invalid Data Type / Invalid Field Name
- ☐ There is no current Index
- ☐ Type mismatch on **<operação>**
- ☐ Variable does not exist
- ☐ Variable is not a **<tipo>**

Erros na execução de Lançamentos Padrões – Função CTRELATION

De forma semelhante ao caso anterior, a maior parte dos erros referentes à função CTRELATION estão relacionados com problemas na configuração dos cadastros dos Relacionamentos dos Lançamentos Padrões. Caso os campos informados na chave de índice deste cadastro ou o número do índice esteja inválido, serão normalmente exibidas as seguintes mensagens de erro:

- ☐ Invalid Field Name
- ☐ Index not found
- ☐ Argument error

5. Processos de Análise

Respeitando as diferenças entre os tipos de erros anteriormente mencionados, podem ser definidos três processos de análise distintos.

Estes processos de análise podem ser generalizados da seguinte forma:

1. Análise somente através do Error.Log ou Análise Simples.
2. Análise através do Error.log com consulta a trechos do código fonte principal ou Análise com Consulta.
3. Análise da situação do sistema no momento do erro ou Análise Complexa.

5.1. Análise Simples

Os erros considerados de Análise Simples são aqueles que não necessitam da consulta ao código fonte, pois todas as informações necessárias podem ser obtidas através da verificação do error.log e das tabelas do sistema.

Os tipos de erros que se enquadram nesta categoria são:

- ☐ **Invalid Field Name**
- ☐ **Incorrect Sintaxe Near (query)**

5.2. Análise com Consulta

Para análise da maioria dos tipos de error.log é necessária uma mínima consulta ao fonte, para identificar:

- ✓ Linha de comando de ocasiona o erro;
- ✓ Informações associadas a esta linha de comando.

Deve-se entender por informações associadas a uma linha de comando:

- ✓ Declaração das variáveis envolvidas nesta linha;
- ✓ Variáveis inicializadas como parâmetros da função;
- ✓ Pontos de Entrada;
- ✓ Consulta a conteúdos de tabelas específicas como SX5 e SX6, através das funções Tabela(Sx5) e GetMV(Sx6);
- ✓ Grupo de perguntas associado, identificado pela função Pergunte().

Através da análise dos fatores acima, que basicamente se referem a inicialização das variáveis ou tabelas envolvidas é normalmente possível determinar os seguintes tipos de erros:

- ☐ **Argument error**
- ☐ **Index not Found**
- ☐ **Invalid Data Type**
- ☐ **Type mismatch on <operador>**
- ☐ **Variable does not exist**
- ☐ **Variable is not a <tipo>**

5.3. Análise Complexa

Nesta categoria enquadram-se os erros que necessitam de uma análise do momento do sistema na situação de erro. Isso implica na necessidade de uma consulta mais abrangente a maioria dos códigos fontes listados na pilha de chamadas e por diversas vezes é necessário o auxílio da ferramenta de Debug do APxIDE.

São considerados erros complexos:

- ☐ **Array out of bounds**
- ☐ **File is in EOF**
- ☐ **There is no current Index**

5.4. Análise de Erros Específicos

Como mencionado anteriormente, são considerados erros específicos aqueles referentes a problemas encontrados na execução de uma Query ou na execução de um Lançamento Padrão.

5.4.1. Análise de Erros na execução de Queries

Os erros relacionados ao tratamento de Queries são visualizados através de mensagens de erros geradas pelos gerenciadores de bancos de dados mas interpretadas pelo sistema e mensagens de erro geradas pela ferramenta TopConnect.

Nestes casos é necessária a análise do erro com o uso de ferramentas complementares, tais como:

- ☐ **Guia de Referência para Mensagens de Erro: ADS, CTREE, BTRIEVE;**
- ☐ **Guia de Referência para Erros do TopConnect;**
- ☐ **Análise do Log de Eventos do TopConnect;**

Ainda nas situações em que o erro está na sintaxe incorreta da query, há a necessidade do conhecimento mínimo da linguagem SQL e da consulta a características específicas dos diversos Bancos de Dados para tratamentos de comandos.

Para estes casos, o melhor procedimento a seguir é:

1º Para mensagens de erros específicas do TopConnect, ADS, CTREE e BTRIEVE, consultar no site do AC o Guia de Referência de Mensagens de Erro.

2º Para erros na estrutura da Query executada, enviar para o suporte juntamente com o arquivo error.log o arquivo contendo os eventos do TopConnect, o qual deve ser gerado através da ferramenta TopManager.

5.4.2. Análise de Erros na execução de Lançamentos Padrões

Erros na execução de Lançamentos Padrões – Função TRANSLCTA

Os erros relacionados à interpretação de conteúdos inválidos no cadastro de lançamentos padrões ocorrem na execução da função TRANSLCTA.

Esta função interpreta o conteúdo de campos específicos dos cadastros SI5 – SIGACON / CT5 – SIGACTB, permitindo a execução de funções para retorno de valores.

Por causa desta funcionalidade, a execução da TRANSLCTA no momento da contabilização depende da integridade das informações dos cadastros dos lançamentos.

Para determinar o lançamento / sequência que ocasionou o erro de execução podem ser observados os seguintes pontos:

1ª Passo: Verificar no arquivo error.log o ALIAS:SI5 ou ALIAS:CT5, determinando em qual lançamento a rotina estava posicionada, e qual o conteúdo dos campos do cadastro.

Em alguns casos o cadastro de lançamentos não está posicionado no lançamento que ocasionou o erro, então é importante avaliar outros aspectos do error.log, mencionados no 2ª passo.

2ª Passo: Normalmente o evento de erro da TRANSLCTA está relacionado com uma das seguintes ocorrências:

- 1. Invalid Field Name**
- 2. Variable does not exist**
- 3. Argument error expected**

2.1. Para as duas primeiras ocorrências devem ser utilizados os procedimentos de análise específicos para as mesmas, o que irá apontar para um conteúdo incorreto de um dos campos do cadastro do lançamento padrão.

2.2. No caso do Argument error expected, no error.log deverá ser localizada a sessão do error.log referente as variáveis em uso pela função TRANSLCTA.

Nesta sessão os conteúdos das variáveis <cConteudo> e <xConteudo> determinarão quais informações definidas no cadastro do lançamento estão incorretas.

Erros na execução de Lançamentos Padrões – Função CTRELATION

A função CTRELATION é executada nos lançamentos padrões, de acordo com o cadastro de Relacionamentos do CTB – Tabela CTL, e seu objetivo é gravar uma chave de pesquisa nas tabelas CTK – campo CTK_KEY e CT2 – campo CT2_KEY.

A chave de pesquisa gravada por esta função permite identificar a origem do lançamento, e é utilizada por diversas funções existentes no módulo CTB (vide documentação sobre Relacionamentos).

A ocorrência de erros durante a execução desta função está relacionada normalmente a duas causas:

1ª Causa: Cadastro incorreto do Relacionamento – Tabela CTL

No cadastro de Relacionamento devem ser informados:

1. Código do Lançamento Padrão;
2. Alias da tabela que contém o gerador do lançamento;
3. Campos que compõe a chave de índice para a busca do gerador do lançamento;
4. Ordem de índice (SIX) correspondente a esta chave, em formato numérico.

O preenchimento incorreto dos campos ORDEM e CHAVE causam erros diversos na execução da CTRELATION, tais como:

1. Index not found;
2. Invalid field name;
3. Variable does not exist.

2ª Causa: Tratamento incorreto da rotina que está efetuando a contabilização

Algumas rotinas do sistema que utilizam queries para montagem de áreas temporárias e otimização de performance fecham os ALIAS das tabelas do sistema.

Neste processo os índices destas tabelas também são fechados, ocasionando o erro: **“There is no current Index”**.

Para esta situação deverá ser avaliada a rotina para determinar a melhor forma de adequá-la para permitir o uso dos relacionamentos no processo de contabilização.

6. Estudo de Casos

6.1. Análise Simples

6.1.1. Invalid Field Name

1. Cabeçalho do erro

Invalid field name in Alias <alias> <nome do campo> on <função> (<fonte>)
<data do fonte> line: <linha>

2. Análise

1. Verificar no error.log a estrutura da tabela ao qual o campo pertence, procurando pelo texto: ALIAS: <tabela>

Esta informação estará disposta no error.log da seguinte forma:

1ª Linha da Estrutura – Informações da Tabela

<caminho do arquivo + nome do arquivo>

Alias: <tabela>;

Filter : <expressão de filtro> ;

Recno: <número do registro posicionado>;

Total Recs: <total de registros na tabela>;

Order: <ordem de índice ativa>

Linhas de Índices da Tabela

Índice (<número do índice>) : <campos da chave do índice>

Linhas de Campos

Campo <numero do campo>: <nome do campo>(<tipo do campo>) :
<conteúdo>

2. Pela análise da estrutura demonstrada no error.log temos:

- ☐ O campo não existe na tabela, pois a estrutura da mesma está divergente da estrutura do SX3, ocasionando o problema.
- ☐ O fonte está referenciando incorretamente um campo que não existe na tabela, ocasionando o problema.

3. Exemplo

```
AP6 Build 7.00.031210A
ERRO THREAD (vmartins, CPD_II) 20/01/2004 07:33:22
invalid field name in Alias SE2->R_E_C_N_O_ on FA340TIT(FINA340.PRX)
28/11/2003 line : 1497

[build:7.00.031210A]
[thread 2308]
Called from FA340COMP(FINA340.PRX) 28/11/2003 line : 349
Called from EXECBROW(MSLIB.PRW) 04/12/2003 line : 534
Called from {|| SETENCH("&COMPENSAR"), ALTERA:=.T.,EXECBROW('FA340COMP',,
3,'SE2',@AENCHO,, 'XFILIAL("SE2")', 'XFILIAL("SE2")')} line : 4470
Called from line : 0
Called from MBRWINDOW(MSLIB.PRW) 04/12/2003 line : 4549
Called from MBROWSE(MSLIB.PRW) 04/12/2003 line : 411
Called from FINA340(FINA340.PRX) 28/11/2003 line : 107
Called from __EXECUTE(APLIB090.PRW) 04/12/2003 line : 971
Called from {|| __EXECUTE('FINA340()', 'XXXXXXXXXX', 'COMPENSAÇÃO CP', '')}
line : 2752
Called from line : 0
Called from SIGAADV(SIGAADV.PRW) 23/04/2002 line : 223

...

** - Alias:SE2; Filter : ; Recno:0; Total Recs:1; Order:0
Campo 1: E2_PREFIXO(C) :CX3
Campo 2: E2_NUM(C) :30101
Campo 3: E2_PARCELA(C) :
Campo 4: E2_TIPO(C) :NF
Campo 5: E2_NATUREZ(C) :11101102
Campo 6: E2_PORTADO(C) :
Campo 7: E2_FORNECE(C) :000503
Campo 8: E2_LOJA(C) :01
Campo 9: E2_NOMFOR(C) :TYCO
Campo 10: E2_EMISSAO(D) :2004/1/20

...

Campo 75: E2_MULTNAT(C) :2
Campo 76: E2_PROJPMS(C) :2
Campo 77: E2_PLLOTE(C) :
Campo 78: E2_CODRET(C) :
Campo 79: E2_DIRF(C) :2
Campo 80: E2_TAXDOL(N) :0
Campo 81: E2_CCONTAB(C) :51207201
Campo 82: E2_CCUSTO(C) :3700
Campo 83: @DELETED@(C) :
```

6.1.2. Incorrect Sintaxe Near (query)

1. Cabeçalho do erro

Incorrect syntax near **<texto>**

<select_parcial> on **<função>** (**<fonte>**) **<data do fonte>** line : **<linha>**

2. Análise

1. Localizar no error.log a variável cQuery, pois a mesma é utilizada como padrão na montagem de queries pelos fontes. Caso contrário, determinar pela análise do fonte qual variável contém a query que ocasionou o erro.
 - o A linha do fonte referenciada no erro normalmente apontará para o conjunto de instruções DbUseArea(...,TcGenQry(**<variável>**)...)
2. Caso a variável cQuery seja encontrada e o conteúdo da variável for igual ao **<select_parcial>** exibido no cabeçalho do erro:
 - o Analisar o texto da query em busca de um caracter " ' " (aspas simples) indevido.
3. Caso a variável cQuery não seja encontrada ou o conteúdo da variável não for igual ao **<select_parcial>** exibido no cabeçalho do erro, deverá ser efetuada uma consulta ao **<fonte>** / **<linha>** / **<data do fonte>** especificados no cabeçalho do erro.
 - o Na linha especificada haverá a chamada da função TcGenQry(.,**<variável>**).
 - o Localizar a **<variável>** no error.log e analisar o texto da query em busca de um caracter " ' " (aspas simples) indevido.

3. Exemplo

```
AP6 Build 7.00.040531P
ERRO THREAD (administrator, KINAWA) 09/07/2004 09:27:24
: TS0170 - [Microsoft][ODBC SQL Server Driver][SQL Server]Line 1: Incorrect
syntax near ' AND E5_NUMERO='.
( From SQLQry_File::SQLQry_File )
SELECT COUNT(*) ESTORNO FROM SE5020 WHERE E5_FILIAL='04' AND E5_PREFIXO=' '
' AND E5_NUMERO='022212' AND E5_PARCELA=' ' AND E5_TIPO='RC ' AND
E5_CLIFOR='000636' AND E5_LOJA='01' AND E5_SEQ='01' AND E5_TIPODOC='ES' AND
E5_DATA <= '20040709' AND D_E_L_E_T_<>'*' on TEMBXCANC(MATXFUNB.PRX)
16/02/2004 line : 4771

[build:7.00.040531P]
[thread 2940]
Called from FA190IMP(FINR190.PRX) 26/08/2003 line : 523
Called from {||LEND| FA190IMP(@LEND,WNREL,CSTRING)}(FINR190.PRX) line : 111
Called from line : 1519
Called from { || EVAL( BACTION, @LSTOP ),LSTOP := .T., ODLG:END()
}(APLIB080.PRW) line : 1519
Called from line : 0
Called from RPTSTATUS(APLIB080.PRW) 04/12/2003 line : 1520
Called from FINR190(FINR190.PRX) 26/08/2003 line : 111
Called from __EXECUTE(APLIB090.PRW) 04/12/2003 line : 971
Called from {|| __EXECUTE('FINR190()','XXXXXXXXXX','RELAÇÃO DE BAIXAS
','')} line : 2752
Called from line : 0
Called from SIGAFIN(SIGAFIN.PRW) 21/08/2001 line : 222

...

TEMBXCANC(MATXFUNB.PRX) 16/02/2004 Param 1: CCHAVE(C) : ' 022212 RC
0006360101
Local 1: CCHAVE(C) : ' 022212 RC 0006360101
Local 2: AAREA(A) :
Local 3: AAREASE5(A) :
Local 4: LRET(L) :.F.
Local 5: CQUERY(C) :SELECT COUNT(*) ESTORNO FROM SE5020 WHERE
E5_FILIAL='04' AND E5_PREFIXO=' ' ' AND E5_NUMERO='022212' AND E5_PARCELA='
' AND E5_TIPO='RC ' AND E5_CLIFOR='000636' AND E5_LOJA='01' AND E5_SEQ='01'
AND E5_TIPODOC='ES' AND E5_DATA <= '20040709' AND D_E_L_E_T_<>'*'
Local 6: CALIAS(C) :__BXCAN
```

Texto da Query

```
SELECT COUNT(*) ESTORNO FROM SE5020 WHERE E5_FILIAL='04' AND E5_PREFIXO=' '
' AND E5_NUMERO='022212' AND E5_PARCELA=' ' AND E5_TIPO='RC ' AND
E5_CLIFOR='000636' AND E5_LOJA='01' AND E5_SEQ='01' AND E5_TIPODOC='ES' AND
E5_DATA <= '20040709' AND D_E_L_E_T_<>'*
```

6.2. Análise com Consulta

6.2.1. Argument error

1. Cabeçalhos do erro

Argument <número do parâmetro> error, expected <tipo> -> <tipo recebido>, function <nome da função> on <rotina>(<fonte>) <data do fonte> line: <linha do fonte>

ou

Argument error on <função/rotina>(<fonte>) <data do fonte> line: <linha do fonte>

2. Análise

1. Verificar no <fonte>/ <data do fonte> / <linha> especificados no cabeçalho do erro quais parâmetros estão sendo informados para a função <nome da função>
2. Verificar no arquivo error.log gerado o conteúdo de cada uma das variáveis de forma a determinar qual está com o conteúdo do <tipo recebido> incorreto informado no cabeçalho do erro.
3. Analisar o <fonte> e/ou <fontes> especificados na Pilha de Funções para determinar:
 - a. Em que momento e com qual conteúdo a variável que ocasionou o erro foi inicializada. (primeira atribuição de conteúdo).
 - b. Em qual/quais momento(s) esta variável sofre alterações de seu conteúdo.
4. Através da análise do item "3." determinar a causa do conteúdo incorreto e proceder na correção.

6.2.2. Index not found

1. Cabeçalho do erro

SetOrder - Index not found. On <função>(<fonte>) <data do fonte> line : <linha>

2. Análise

1. Verificar no <fonte>/ <data do fonte> / <linha> especificados no cabeçalho do erro qual número de índice está sendo utilizado na chamada da função DbSetOrder(<numero_do_índice>).
2. Verificar através da análise do <fonte> ou do error.log qual a <tabela> ativa no momento da execução da função DbSetOrder.
 - o Pela análise do fonte deve ser verificada a última ocorrência da função DbSelectArea("<alias>") ou verificar se a sintaxe utilizada é: <alias> -> (DbSelectArea("<alias>")).
 - o Pela análise do error.log, deve ser localizado o <alias> indicado pelos símbolos ** no início da 1ª Linha das Informações da Tabela.

1ª Linha – Informações da Tabela

Alias Ativo: <**>: somente estará marcando o ALIAS ativo no momento do erro
<caminho do arquivo + nome do arquivo>
Alias: <tabela>;
Filter : <expressão de filtro>;
Recno: <número do registro posicionado>;
Total Recs: <total de registros na tabela>;
Order: <ordem de índice ativa>

3. Exemplo

```
AP7 Build 7.00.031210A
ERRO THREAD (ansantos, DIGITACAO) 30/12/2003 11:19:05
: SetOrder - Index not found. on CTRELATION(MATXFUNC.PRX) 12/09/2003 line :
6231
```

Trechos do arquivo error.log

```
CTRELATION(MATXFUNC.PRX) 12/09/2003 Param 1: CPADRAO(C) : 610
  Local 1: CPADRAO(C) :610
  Local 2: ASAVEAREA(A) :
  Local 3: C_ALIAS(C) :SD2
  Local 4: C_CHAVE(C)
: D2_FILIAL+D2_DOC+D2_SERIE+D2_CLIENTE+D2_LOJA+D2_COD+D2_ITEM
  Local 5: N_ORDEM(N) :3
  Local 6: CCHAVEBUSCA(C) :
  Local 7: AAREA(A) :

...

CTL010 - Alias:CTL; Filter : ; Recno:18; Total Recs:22; Order:1
  Indice (0) : **CTL0101 CTL_FILIAL+CTL_LP
  Campo 1: CTL_FILIAL(C) :01
  Campo 2: CTL_LP(C) :610
  Campo 3: CTL_KEY(C) ...
  Campo 4: CTL_ORDER(C) :3
  Campo 5: CTL_DESC(C) :DOCUMENTO DE SAIDA (CABECALHO)
  Campo 6: CTL_EXECUT(C) :
  Campo 7: CTL_ALIAS(C) :SD2

...

SD1010 - Alias:SD1; Filter : ; Recno:11091; Total Recs:6091; Order:1
  Indice (0) : **SD10101
  Indice (1) : SD10102
  Indice (2) : SD10103
  Indice (3) : SD10104
  Indice (4) : SD10105
  Indice (5) : SD10106
  Indice (6) : SD10107
  Indice (7) : SD10108
  Indice (8) : SD10109
  Indice (9) : SD1010A
  Campo 1: ...

...

** - Alias:SD2; Filter : ; Recno:1; Total Recs:1; Order:0
  Campo 1: SF2RECNO(N) :260
  Campo 2: SA1RECNO(N) :1
  Campo 3: SF4RECNO(N) :13
  Campo 4: SB1RECNO(N) :75
```

6.2.3. Invalid Data Type

1. Cabeçalho do erro

Invalid data type on <campo> in file <lib de tecnologia> at line <linha da lib> on <função/rotina>(<fonte>) <data do fonte> line : <linha do fonte>

2. Análise

1. Verificar se o tipo do campo no SX3 e da tabela/arquivo estão coerentes e de acordo com o padrão ou documentações fornecidas pela Microsiga.
2. Verificar no <fonte> / <data do fonte> / <linha> especificados no cabeçalho do erro qual conteúdo é atribuído ao <campo> também especificado no cabeçalho. Este conteúdo pode ser:
 - a. Uma variável ou uma operação entre variáveis utilizadas pela rotina;
 - b. Retorno de uma outra função;
 - c. Valores fixos especificados na rotina;
 - d. Formato incorreto dos dados em uma tabela do sistema.
3. Dependendo do tipo de conteúdo atribuído ao campo:
 - a. **Variável ou Operação entre variáveis:** Verificar no arquivo error.log gerado o conteúdo das variáveis de forma a determinar o que está causando um retorno com um tipo diferente do esperado pelo campo.
 - b. **Retorno de outra função:** Determinar através da análise do fonte da função o que está causando um retorno com um tipo diferente do esperado pelo campo.
 - c. **Valores fixos:** Verificar se o tipo de informação atribuída pela rotina está coerente com o tipo do campo.
4. Através da análise do item "3." determinar a causa da atribuição incorreta e proceder na correção.

3. Exemplo

```
AP6 Build 7.00.031002A
ERRO THREAD (sigal484, CLAUDIACABRAL) 03/12/2003 11:57:53
Invalid data type on (CTC_DATA) in file C:\ADVTEC7\advpl\instrvar.inl at
line 337 on GRAVACTC(CTBXFUN.PRX) 08/10/2003 line : 457
```

6.2.4. Type mismatch on <operation>

1. Cabeçalho do erro

Type mismatch on <operação> on <função/rotina>() <data do fonte>
line: <linha do fonte>

2. Análise

1. Verificar no / <data do fonte> / <linha> especificados no cabeçalho do erro quais variáveis estão sendo utilizadas na <operação>
2. Verificar no arquivo error.log gerado o conteúdo de cada uma das variáveis de forma a determinar qual está com o conteúdo inválido para o tipo de <operação> empregado.
3. Analisar o e/ou <fontes> especificados na Pilha de Funções para determinar:
 - a. Em que momento e com qual conteúdo a variável que ocasionou o erro foi inicializada. (primeira atribuição de conteúdo).
 - b. Em qual/quais momento(s) esta variável sofre alterações de seu conteúdo.
4. Através da análise do item "3." determinar a causa do conteúdo incorreto e proceder na correção.

3. Exemplo

```
AP7 Build 7.00.040531P
ERRO THREAD (Administrador, INFO) 05/07/2004 16:48:49
type mismatch on compare on FA190IMP(FINR190.PRX) 26/08/2003 line : 535
```

Trecho do Fonte FINR190 - FA190IMP

```
If NEWSE5->E5_FILORIG < mv_par33 .or. NEWSE5->E5_FILORIG > mv_par34
    dbSelectArea("NEWSE5")
    NEWSE5->( dbSkip() )
    Loop
Endif
```

Trechos do arquivo error.log

```
Variaveis em uso
Publicas
...
```

```
Publica 133: MV_PAR33(D) :1997/1/1
Publica 134: MV_PAR34(D) :2009/12/31
```

```
** - Alias:NEWSE5; Filter : ; Recno:1; Total Recs:1; Order:0
```

```
...
```

```
Campo 45: E5_FILORIG(C) :
```

6.2.5. Variable does not exist

1. Cabeçalho do erro

Variable does not exist <variável> on <função/rotina>(<fonte>) <data do fonte>
line: <linha do fonte>

2. Análise

1. Verificar no error.log se <variável> especificada no cabeçalho do erro existe em alguma das funções registradas na Pilha de Funções.
 - a. Caso a variável exista, deve ser observada a seguinte regra referente ao escopo:
 - ❑ Escopo Local: deve estar contida na listagem das variáveis da <função/rotina> especificada no cabeçalho do erro.
 - ❑ Escopo Private: deve estar contida na listagem das variáveis de uma <função/rotina> que foi responsável pela execução da <função/rotina> especificada no cabeçalho do erro. Esta análise somente é possível através da verificação das chamadas pela Pilha de funções.
 - ❑ Escopo Public: Esta variável está disponível para todas as funções especificadas na Pilha de Funções.
 - b. Caso o escopo da variável não esteja condizente com sua utilização na função, proceder na correção.
2. Se a verificação do item "1." indicou que a variável não existe no error.log, devem ser observados:
 - a. Grafia da variável: A variável é utilizada com um nome similar em outros pontos da rotina, caracterizando erro de codificação.
 - b. Falta de Declaração: A grafia da variável está correta, caracterizando falta da declaração da mesma.
 - c. Campo sem ALIAS: A tabela ao qual pertence um campo não é o ALIAS ativo no momento da verificação do mesmo e a sintaxe utilizada na utilização do campo não for ALIAS->CAMPO, caracterizando erro de codificação.
 - d. Utilização Indevida: A variável é normalmente empregada em outros processos do sistema, mas não foi disponibilizada para a <rotina /função>, caracterizando erro de codificação.
3. Através da análise do item "2." proceder na correção do erro caracterizado.

3. Exemplo

```
AP6 Build 7.00.040831P  
ERRO THREAD (janisse, DP4) 20/09/2004 16:32:20  
variable does not exist CMODRETPIS on FVERMINIMP(FINA050.PRX) 14/09/2004  
line : 8755
```

Trechos do arquivo error.log

```
FVERMINIMP(FINA050.PRX) 14/09/2004 Param 1: NVALOR(N) : 63.46  
Private 1: NPISORI(N) : 0  
Private 2: NCOFORI(N) : 0  
Private 3: NCSLORI(N) : 0  
Local 1: NVALOR(N) : 63.46  
Local 2: NVLMINIMP(N) : 5000  
Local 3: NCOND(N) : 0  
Local 4: NVALSEST(N) : 0  
Local 5: LCONTRRET(L) : .T.
```

Nota: Pela definição de escopos de variáveis, se CMODRETPIS existisse definida como tipo PRIVATE em uma função executada anteriormente na pilha, esta definição a tornaria válida na função FVERMINIMP.

Trecho do Fonte FINA050 - FVERMINIMP

```
Function FVerMinImp(nValor)  
Local nVlMinImp := GetNewPar("MV_VL10925", 5000)  
Local nCond := 0  
Local nValSest := ...  
Local lContrRet := ...  
  
If lAltera  
    lAlterNat := .T.  
Endif  
  
DEFAULT nValor := M->E2_VALOR  
  
nPisOri := IIf(Type("nPisOri") != "N" , 0, nPisOri)  
nCofOri := IIf(Type("nCofOri") != "N" , 0, nCofOri)  
nCslOri := IIf(Type("nCslOri") != "N" , 0, nCslOri)  
  
If CposImpPagar() .and. lContrRet  
    //Nao retem Pis,Cofins,CSLL  
    If cModRetPis == "3" //Nao retem PIS  
        nVlRetPis := M->E2_PIS  
        nVlRetCof := M->E2_COFINS  
        nVlRetCsl := M->E2_CSLL  
        M->E2_VALOR += M->E2_PIS + M->E2_COFINS + M->E2_CSLL  
        M->E2_PIS := 0  
        M->E2_COFINS := 0  
        M->E2_CSLL := 0  
    Else  
        ...
```

6.2.6. Variable is not <tipo>

1. Cabeçalho do erro

Variable is not <tipo> on <função/rotina>(<fonte>) <data do fonte> line: <linha do fonte>

2. Análise

1. Verificar no <fonte>/ <data do fonte> / <linha> especificados no cabeçalho do erro quais variáveis estão sendo utilizadas na <função/rotina>
2. Verificar no arquivo error.log gerado o conteúdo de cada uma das variáveis de forma a determinar qual está com o conteúdo inválido para o <tipo> esperado pela <função/rotina>
3. Analisar o <fonte> e/ou <fontes> especificados na Pilha de Funções para determinar:
 - a. Em que momento e com qual conteúdo a variável que ocasionou o erro foi inicializada. (primeira atribuição de conteúdo).
 - b. Em qual/quais momento(s) esta variável sofre alterações de seu conteúdo.
4. Através da análise do item "3." determinar a causa do conteúdo incorreto e proceder na correção.

3. Exemplo

```
AP7 Build 7.00.040831P
ERRO THREAD (3rg24, MARINHEIRO) 19/10/2004 12:58:54
variable is not array on SALDOTIT(FINXFUN.PRX) 16/09/2004 line : 143
```

Trecho do Fonte FINXFUN – SALDOTIT

```
If ExistProc( cProcedure ) .and. ( TcSrvType() <> "AS/400" )
    aResult := {}

    ...

    aResult := TCSPEXEC( ... )

    ...

    nSaldo := aResult[1]
```

Trechos do arquivo error.log

```
SALDOTIT(FINXFUN.PRX) 16/09/2004 Param 1: CPREFIXO(C) :
Param 2: CNUMERO(C) : 000002
Param 3: CPARCELA(C) :

...

Private 1: ARESULT(U) :NIL
```

6.3. Dicas para Análise de Erros Complexos

6.3.1. Array out of bounds

1. Cabeçalho do erro

Array out of bounds on <função/rotina> (<fonte>) <data do fonte> line: <linha do fonte>

2. Dicas

1. Array com conteúdo incompleto:

- Verificar através da análise do fluxo do programa quais dados o array deveria conter em uma situação normal.
- Através desta análise, determinar as possíveis causas para que estes dados não estejam presentes na estrutura do array e proceder com as correções necessárias.

Exemplo:

```
aCampos := { }
```

```
If cAlias := "001"
```

```
    AADD(aCampos,"001_CAMPO01")
```

```
    AADD(aCampos,"001_CAMPO02")
```

```
    AADD(aCampos,"001_CAMPO03")
```

```
Elseif cAlias := "002"
```

```
    AADD(aCampos,"002_CAMPO01")
```

```
    AADD(aCampos,"002_CAMPO02")
```

```
    AADD(aCampos,"002_CAMPO03")
```

```
Elseif cAlias := "003"
```

```
    AADD(aCampos,"003_CAMPO01")
```

```
    AADD(aCampos,"003_CAMPO02")
```

```
Endif
```

```
cCampo := aCampos[3]
```


2. Limites do array excedidos:

- a. Verificar através da análise do fluxo do programa quais as dimensões normais do array.
- b. Verificar através da análise das variáveis do error.log se os limites do array foram desrespeitados por um conteúdo incorreto em um dos indexadores do utilizados no mesmo.

Exemplo: Array: aValores – Indexador “i”

```
aValores := Array(9)
```

```
For i:= 1 to 10 // Excede o tamanho do array
```

```
    nSoma += aValores[i]
```

```
Next i
```

6.3.2. File is in EOF

1. Cabeçalho do erro

<Operação/Processo> - File is in EOF - <tabela/arquivo> in file <lib de tecnologia> at line <linha da lib> on <função/rotina>(<fonte>) <data do fonte> line: <linha do fonte>

2. Dicas

1. Alias posicionado incorretamente pela rotina:

- a. Efetuar a análise do fluxo da rotina para determinar em qual momento o ALIAS utilizado na operação que acusou o erro deveria ter sido posicionado.
- b. Dependendo da análise do fluxo da rotina, podem ser constatadas as seguintes situações:
 - i. A rotina não efetuou o tratamento para posicionar o alias;
 - ii. O tratamento estava condicionado a uma instrução e não foi executada devido à parametrização da rotina;
 - iii. O tratamento foi executado, mas a instrução recebeu um conjunto de parâmetros incorretos, desposicionando o arquivo.
- c. A análise do fluxo do programa depende do conteúdo das variáveis contido no error.log

2. Alias desposicionado por um tratamento indevido de uma customização:

- a. Uma customização ou instrução específica do ambiente do cliente causou o desposicionamento do ALIAS necessário à rotina padrão do sistema, em uma das situações abaixo:
 - i. A customização não armazenou antes de sua execução e restaurou ao final da mesma o ambiente do sistema, utilizando as funções GetArea() e RestArea();
 - ii. A customização manipulou incorretamente o conteúdo de variáveis do tipo PRIVATE utilizada pela rotina padrão do sistema.
- b. Estas situações normalmente ocorrem no emprego do seguintes recursos do sistema:
 - i. Pontos de entrada;
 - ii. Funções específicas em Lançamentos Padrões ou a utilização da função Posicione();
 - iii. Funções específicas em gatilhos ou validações de campos.

6.3.3. There is no current Index

1. Cabeçalho do erro

<Arquivo/Tabela>: There is no current index on <função/rotina>(<fonte>) <data do fonte> line: <linha do fonte>

2. Dicas

1. Utilização incorreta da função dbCloseArea() / dbCloseIndex()

- a. A rotina padrão ou uma rotina específica utilizou a função dbCloseArea() ou dbCloseIndex() mas não restaurou o ALIAS ou os índices ao término de sua execução ou na mudança de processos.

2. Falha na execução da função IndRegua()

- a. Na execução da função IndRegua() ocorreu um erro que impossibilitou a criação do índice temporário utilizado pela rotina, ocasionando o erro.
- b. Como o erro ocorreu quando a rotina tentou utilizar o índice específico, devem ser analisados os erros nos gerenciadores de bancos (TopConnect) e/ou gerenciadores de arquivos (ADS/CTREE entre outros).

3. Tratamento incorreto para o ambiente TopConnect

- a. Para ambiente TopConnect a função dbSetIndex() não deve ser utilizada, pois referencia um arquivo temporário local (\SIGAADV\), sendo que neste ambiente os arquivos de índices temporários para tabelas que estão no Banco de Dados também serão criados no mesmo.

7. Casos de Análise

7.1. Caso de Análise Simples

Cabeçalho do erro

```
AP7 Build 7.00.040308A
ERRO THREAD (borba, TML191) 13/05/2004 12:42:43
: TS0170 - [Microsoft][ODBC SQL Server Driver][SQL Server]
Line 1: Incorrect syntax near 'RA'.
( From SQLQry_File::SQLQry_File )
SELECT COUNT(*) ESTORNO FROM SE5010 WHERE E5_FILIAL='01' AND
E5_PREFIXO='ADI' AND E5_NUMERO='PV2055' AND E5_PARCELA=' ' AND E5_TIPO='RA
' AND E5_CLIFOR='C836 ' AND E5_LOJA='02' AND E5_SEQ=' ' AND E5_TIPODOC='ES'
AND E5_DATA <= '20040513' AND D_E_L_E_T_<>'*'
on TEMBXCANC(MATXFUNB.PRX) 03/03/2004 line : 4754
```

1º Passo: Interpretação do erro

Incorrect syntax near **<texto>**
<select_parcial> on **<função>** (**<fonte>**) **<data do fonte>** line : **<linha>**

Aonde:

<texto>	'RA'
<select_parcial>	SELECT COUNT(*) ESTORNO FROM SE5010 WHERE E5_FILIAL='01' AND E5_PREFIXO='ADI' AND E5_NUMERO='PV2055' AND E5_PARCELA=' ' AND E5_TIPO='RA ' AND E5_CLIFOR='C836 ' AND E5_LOJA='02' AND E5_SEQ=' ' AND E5_TIPODOC='ES' AND E5_DATA <= '20040513' AND D_E_L_E_T_<>'*'
<função>	TEMBXCANC
<fonte>	MATXFUNB.PRX
<data do fonte>	03/03/2004
<linha>	4754

2º Passo: Análise do Error.log

1. Localizar a variável que contém a query completa (normalmente cQuery) na função na qual ocorreu o erro:

Trecho do arquivo error.log

```
TEMBXCANC(MATXFUNB.PRX) 03/03/2004 Param 1: CCHAVE(C) : ADIPV2055'RA
C836 02
Local 1: CCHAVE(C) :ADIPV2055'RA C836 02
Local 2: AAREA(A) :
Local 3: AAREASE5(A) :
Local 4: LRET(L) :.F.
Local 5: CQUERY(C) :SELECT COUNT(*) ESTORNO FROM SE5010 WHERE
E5_FILIAL='01' AND E5_PREFIXO='ADI' AND E5_NUMERO='PV2055' AND
E5_PARCELA=' ' AND E5_TIPO='RA ' AND E5_CLIFOR='C836 ' AND E5_LOJA='02'
AND E5_SEQ=' ' AND E5_TIPODOC='ES' AND E5_DATA <= '20040513' AND
D_E_L_E_T_<>'*'
Local 6: CALIAS(C) :__BXCAN
```

3º Passo: Análise do conteúdo da variável

1. Conteúdo da variável cQuery alinhado:

```
SELECT COUNT(*) ESTORNO FROM SE5010
WHERE
E5_FILIAL='01' AND
E5_PREFIXO='ADI' AND
E5_NUMERO='PV2055' AND
E5_PARCELA=' ' AND
E5_TIPO='RA ' AND
E5_CLIFOR='C836 ' AND
E5_LOJA='02' AND
E5_SEQ=' ' AND
E5_TIPODOC='ES' AND
E5_DATA <= '20040513' AND
D_E_L_E_T_<>'*'
```

2. Análise da query para identificar a string que contém o caractere " ' " (apóstrofe):

```
E5_PARCELA=' ' AND
```

4º Passo: Interpretação da Análise e Solução

1. Os campos das tabelas do sistema não podem conter o caracter apóstrofe (') pois o mesmo é interpretado pelos bancos de dados padrão SQL.
2. Deve ser removido o caracter inválido do registro do SE5 indicado na query e do registro do SE1 (RA – Recebimento Antecipado). Também devem ser avaliadas outras ocorrências desta situação na base.

7.2. Caso de Análise com Consulta

Cabeçalho do erro

MP8 Build 7.00.041103A
ERRO THREAD (microsiga, BRAS064) 13/12/2004 18:13:47
type mismatch on \$ on FA050TIPO(FINA050.PRX) 25/11/2004 line : 1227

1º Passo: Verificar a linha do fonte

1. Trecho do fonte FINA050.PRX – 25/11/2004:

Linhas	Código
1227	If !Retorna .And. m->e2_naturez\$\$(GetMv("MV_INSS")) .And. !m->e2_tipo \$ MVINSS
1228	Help(" ",1,"E2_TIPO")
1229	!Retorna := .F.
1230	EndIf

2. Instruções que podem ter ocasionado o erro:

- ☐ m->e2_naturez \$ \$(GetMv("MV_INSS"))
- ☐ !m->e2_tipo \$ MVINSS

2º Passo: Avaliar conteúdo das variáveis envolvidas na operação

1. Variáveis da instrução: m->e2_naturez \$ \$(GetMv("MV_INSS"))

Variável	Conteúdo
m->e2_naturez	"branco / vazio"
MV_INSS	26003

1.1. Origem do conteúdo da variável: m->e2_naturez

- ☐ Conteúdo disponível apenas no error.log, para a variável PRIVATE E2_NATUREZ, identificada pelo uso do ALIAS "M->" (variáveis em memória para utilização em telas).

Error.log	Conteúdo
716	AXINCLUI(MATXATU.PRX) 28/10/2004 Param 1: CALIAS(C) : SE2
717	Param 2: NREG(N) : 5000
718	Param 3: NOPC(N) : 3
...	...
730	Private 1: ATELA(A) :
731	Private 2: AGETS(A) :
...	...
737	Private 8: E2_NATUREZ(C) :

1.2. Origem do conteúdo da variável: **MV_INSS**

- ❑ Conteúdo disponível na tabela SX6 da empresa/filial corrente ou no error.log, caso o ALIAS SX6 esteja posicionado.

Error.log	Conteúdo
1092	e:\protheus8\protheus_data\systemproducao\sx6020.dbf - Alias: SX6; Filter : ;
1093	Recno: 874; Total Recs: 2107; Order: 1
1094	Indice (0) : **SX60201 X6_FIL+X6_VAR
1095	Campo 1: X6_FIL(C) :
1096	Campo 2: X6_VAR(C) : MV_INSS
1096	Campo 3: X6_TIPO(C) : C
...	...
1106	Campo 13: X6_CONTEUD(C) : 26003

2. Variáveis da instrução: **m->e2_tipo \$ MVINSS**

Variável	Conteúdo
m->e2_tipo	"DP"
MVINSS	"INS"

2.1 Origem do conteúdo da variável: **m->e2_tipo**

- ❑ Conteúdo disponível apenas no error.log, para a variável PRIVATE E2_TIPO, identificada pelo uso do ALIAS "M->" (variáveis em memória para utilização em telas).

Error.log	Conteúdo
716	AXINCLUI(MATXATU.PRX) 28/10/2004 Param 1: CALIAS(C) : SE2
717	Param 2: NREG(N) : 5000
718	Param 3: NOPC(N) : 3
...	...
730	Private 1: ATELA(A) :
731	Private 2: AGETS(A) :
...	...
736	Private 7: E2_TIPO(C) : DP

2.2 Origem do conteúdo da variável: **MVINSS**

- ❑ Conteúdo disponível apenas no error.log, para a variável PUBLICA MVINSS

Error.log	Conteúdo
30	Variaveis em uso
31	Publicas
32	Publica 1: OAPP(O) : O
...	...
201	Publica 170: MVTAXA(C) : TX
202	Publica 171: MVTXA(C) : TXA
203	Publica 172: MVIRF(C) : IRF
204	Publica 173: MVINSS(C) : INS

3º Passo: Determinar a operação que manipulou um conteúdo incorreto

1. Instrução: **m->e2_naturez \$ &(GetMv("MV_INSS"))**

1.1. Interpretação da instrução com os conteúdos:

"branco" \$ &(26003)

"branco" \$ 26003

1.2. Tipos das variáveis envolvidas

☒ Caractere \$ Numérico

2. Instrução: **m->e2_tipo \$ MVINSS**

2.1. Interpretação da instrução com os conteúdos:

"DP" \$ "INS"

2.2. Tipos das variáveis envolvidas

☒ Caractere \$ Caractere

4º Passo: Interpretação da Análise e Solução

1. O conteúdo do parâmetro MV_INSS está incorreto, pois o mesmo será utilizado em uma operação com & (macro).
2. Deverá ser alterado o conteúdo de **26003** para **"26003"** permitindo a correta interpretação do parâmetro.

7.3. Caso de Análise Complexa

Cabeçalho do erro

AP7 Build 7.00.040308A
ERRO THREAD (ventura, CPDX) 22/10/2004 10:19:31
array out of bounds on af060grvmov(ATFA060.PRX) 20/10/2004 line : 2345

1º Passo: Verificar a linha do fonte

1. Trecho do fonte ATFA060.PRX – 20.10.2004

Linhas	Código
2344	If cOcorren == "03" // Origem
2345	SN4 -> N4_CCUSTO := &("SN3->" + aCpDigit[2][1])
2346	If Len(aCpDigit) >= 3
2347	SN4 -> N4_SUBCTA := &("SN3->" + aCpDigit[3][1])
2348	Endif
2349	If Len(aCpDigit) >= 4
2350	SN4 -> N4_CLVL := &("SN3->" + aCpDigit[4][1])
2351	Endif
2352	Else // Destino
2353	SN4 -> N4_CCUSTO := CUSTBEMCTB
2354	SN4 -> N4_SUBCTA := SUBCCONCTB
2355	SN4 -> N4_CLVL := CLVLCONCTB
	Endif

2. Instruções que podem ter ocasionado o erro:

❑ SN4 -> N4_CCUSTO := &("SN3->" + aCpDigit[2][1])

2º Passo: Determinar as operações que atribuíram o conteúdo ao array

1. Funções que deverão ser avaliadas, de acordo com o fluxo do error.log:

Error.log	Conteúdo
8	Called from AF060GRVFIL(ATFA060.PRX) 20/10/2004 line : 1663
9	Called from AF060TRANS(ATFA060.PRX) 20/10/2004 line : 558
10	Called from EXECBROW(MSLIB.PRW) 28/07/2004 line : 647
11	Called from { SETENCH("&TRANSFERIR", ALTERA:=T.,EXECBROW('AF060TRANS', 3,'SN3',@AENCHO,,XFILIAL("SN3"),XFILIAL("SN3"),.F.,T.)) line : 4721
12	Called from line : 0
13	Called from MBRWINDOW(MSLIB.PRW) 28/07/2004 line : 4839
14	Called from MBROWSE(MSLIB.PRW) 28/07/2004 line : 435
15	Called from ATFA060(ATFA060.PRX) 20/10/2004 line : 55
16	Called from __EXECUTE(APLIB090.PRW) 28/07/2004 line : 719
17	Called from __EXECUTE('ATFA060()','XXXXXXXXXX','TRANSFERÊNCIAS','01','&ATUALIZAçõES- >MOVIMENTOS',1)) line : 2343
18	Called from line : 0
19	Called from MSAPP:RUNAPP(APLIB000.PRW) 28/07/2004 line : 352
20	Called from SIGAADV(APLIB000.PRW) 28/07/2004 line : 734

1.1. Lista das funções em ordem de avaliação:

- ☐ ATFA060
- ☐ AF060TRANS
- ☐ AF060GRVFIL
- ☐ AF060GRVMOV (cabeçalho do erro)

2. Trechos do fonte ATFA060.PTX – 20.10.2004:

1º Ponto – AF060TRANS	
Linhas	Código
109	Local aVar := { { "", "", "", "" } } // Somente inicializo
110	Local aCpDigit := { { "N3_CCNTAB", "N3_CCORREC", "N3_CDEPREC", "N3_CCDEPR", "N3_CDESP", "N3_CCUSTO", "SN1->N1_LOCAL" } }
111	Local oLocal, oCCusto

2º Ponto – AF060TRANS	
Linhas	Código
185	If ICc
186	Aadd(aTitFolder, CtbSayApro("CTT"))
187	Aadd(aObjetos, { ,,,"CTT", Bcc })
188	Aadd(aVar, Aclone(aVar[1]))
189	Aadd(aCpDigit, { "N3_CUSTBEM", "N3_CCCORR", "N3_CCDESP", "N3_CCCDEP", "N3_CCCDES" })
190	Endif

3º Ponto – AF060TRANS	
Linhas	Código
192	If lItem
193	Aadd(aTitFolder, CtbSayApro("CTD"))
194	Aadd(aObjetos, { ,,,"CTD", bItem })
195	Aadd(aVar, Aclone(aVar[1]))
196	Aadd(aCpDigit, { "N3_SUBCCON", "N3_SUBCCOR", "N3_SUBCDEP", "N3_SUBCCDE", "N3_SUBCDES" })
197	Endif

4º Ponto – AF060TRANS	
Linhas	Código
199	If ICIVl
200	Aadd(aTitFolder, CtbSayApro("CTH"))
201	Aadd(aObjetos, { ,,,"CTH", bCIVl })
202	Aadd(aVar, Aclone(aVar[1]))
203	Aadd(aCpDigit, { "N3_CLVLCON", "N3_CLVLCOR", "N3_CLVLDEP", "N3_CLVLCDE", "N3_CLVLDES" })
204	Endif

3. Conteúdo esperado do array para cada um dos pontos avaliados:

Array aCpDigit			
01		U	
01	01	C	N3_CCNTAB
01	02	C	N3_CCORREC
01	03	C	N3_CDEPREC
01	04	C	N3_CCDEPR
01	05	C	N3_CDESP

Array aCpDigit			
01	06	C	N3_CCUSTO
01	07	C	N1_LOCAL
02		U	
02	01	C	N3_CUSTBEM
02	02	C	N3_CCCORR
02	03	C	N3_CCDESP
02	04	C	N3_CCCDEP
02	05	C	N3_CCCDES
03		U	
03	01	C	N3_SUBCCON
03	02	C	N3_SUBCCOR
03	03	C	N3_SUBCDEP
03	04	C	N3_SUBCCDE
03	05	C	N3_SUBCDES
04		U	
04	01	C	N3_CLVLCON
04	02	C	N3_CLVLCOR
04	03	C	N3_CLVLDEP
04	04	C	N3_CLVLCDE
04	05	C	N3_CLVLDES

3º Passo: Determinar as possíveis situações de falha em cada operação

1. Pela avaliação dos 4 pontos do fonte ATFA060.PRX nos quais o conteúdo do array aCpDigit é atribuído, os pontos nos quais podem ocorrer falhas são:

- 1.1. **2º Ponto** : variável ICC
- 1.2. **3º Ponto** : variável IItem
- 1.3. **4º Ponto** : variável IClvl

Motivo 01: A atribuição do conteúdo do array está condicionada a estas variáveis. Caso o conteúdo de alguma dessas seja .F. (falso) não serão adicionadas as posições 02, 03 e 04.

Motivo 02: Conforme trecho do fonte ATFA060.PRX entre as linhas 2344 e 2351 estas variáveis não são consideradas na utilização do array aCpDigit.

2. Avaliação das possíveis situações de falha encontradas:

- 2.1. **2º Ponto** : variável ICC – conteúdo: .F. (falso)

Error.log	Conteúdo
519 520 521 571	AF060TRANS(ATFA060.PRX) 20/10/2004 Param 1: CALIAS(C) : SN3 Param 2: NREG(N) : 18806 Param 3: NOPC(N) : 3 ... Local 29: LCC(L) :.F.

- 2.2. **3º Ponto** : variável IItem – conteúdo: .F. (falso)

Error.log	Conteúdo
519 520 521 572	AF060TRANS(ATFA060.PRX) 20/10/2004 Param 1: CALIAS(C) : SN3 Param 2: NREG(N) : 18806 Param 3: NOPC(N) : 3 ... Local 30: LITEM(L) :.F.

2.3. **4º Ponto** : variável ICIVL – conteúdo: .F. (falso)

Error.log	Conteúdo
519	AF060TRANS(ATFA060.PRX) 20/10/2004 Param 1: CALIAS(C) : SN3 Param 2: NREG(N) : 18806 Param 3: NOPC(N) : 3 ... Local 31: LCLVL(L) :.F.
520	
521	
573	

4º Passo: Interpretação da Análise e Próximas Ações

1. Pela análise do conteúdo das variáveis nos pontos determinados como possíveis causas do problema estava coerente com a situação de erro, contatou-se a causa do problema:

Causa do Problema: Foi utilizada uma posição do array aCpDigit que não existe, pois não foi executada a atribuição de conteúdo do 2º passo, que criaria aCpDigit[2][1], utilizado na linha 2345 – fonte ATFA060.PRX

2. Pela análise do fonte, constatou-se que é um problema no controle do array, pois na atribuição são utilizadas variáveis de controle, as quais não estão presentes quando o conteúdo do array é acessado.

Motivo: O fonte não efetua o tratamento correto para acessar o conteúdo do array aCpDigit no trecho entre as linhas 2344 e 2351 – fonte ATFA060.PRX

3. Como o “Motivo” aponta o fonte como causador do erro, temos:

Solução / Próximos passos: Deverá ser avaliada a versão mais recente do fonte ATFA060.PRX, referente ao tratamento do array aCpDigit na função AF060GRVMOV. Caso o fonte atual já possua o controle adequado, deverá ser aplicada uma patch do mesmo no RPO no qual ocorre a situação de erro, caso contrário deverá ser aberta uma solicitação de correção de erro para o fonte.

7.4. Caso de Análise Específica: Lançamentos Padrões

Cabeçalho do erro

MP8 Build 7.00.040608A
ERRO THREAD (admin, SERVIDOR) 20/07/2004 09:39:21
variable does not exist PF on TRANSLCTA(TRANSLCT.PRG) 09/01/2004 line : 35

1º Passo: Determinar o lançamento padrão em execução

1. Avaliar pela pilha de chamada de funções a rotina que originou o processo de contabilização. A rotina é caracterizada por executar a função DETPROVA:

Error.log	Conteúdo
8	Called from DETPROVA(MATXFUNA.PRX) 22/03/2004 line : 6208
9	Called from FA370PROCESSA(FINA370.PRX) 05/02/2004 line : 1160
10	Called from { [LEND] FA370PROCESSA() }(FINA370.PRX) line : 105
11	Called from line : 43
12	Called from { EVAL(BACTION, @LEND),LEND := .T. , ODLG:END() }(MSPROCES.PRW) line : 43
13	Called from line : 0
14	Called from PROCESSA(MSPROCES.PRW) 10/01/2004 line : 45
15	Called from FINA370(FINA370.PRX) 05/02/2004 line : 105
16	Called from __EXECUTE(APLIB090.PRW) 12/05/2004 line : 754
17	Called from MDIEXECUTE(APLIB260.PRW) 03/05/2004 line : 729

2. Avaliar pelo error.log (variável cPadrão) ou através da análise do fonte / rotina, qual o lançamento padrão executado:

Error.log	Conteúdo
380	FA370PROCESSA(FINA370.PRX) 05/02/2004 Param 1: LBAT(L) : .F.
381	Private 1: CCAMPO(C) :E5_DATA
382	Private 2: INCLUI(L) :.T.
...	...
392	Local 7: CPADRAO(C) :510

3. Avaliar pelo error.log se o ALIAS SI5/CT5 está posicionado no lançamento padrão em execução, e se na estrutura deste lançamento há indicação da variável apontada no erro:

Error.log	Conteúdo
1510	d:\protheus8\teste_data\dadosadv\ct5010.dbf - Alias:CT5; Filter : ; Recno:218; Total Recs:374; Order:1
1511	Indice (0) : **CT50101 CT5 FILIAL+CT5_LANPAD+CT5_SEQUEN
1512	Campo 1: CT5_FILIAL(C) :PF
1513	Campo 2: CT5_LANPAD(C) :515
1514	Campo 3: CT5_SEQUEN(C) :101
1515	Campo 4: CT5_DESC(C) :CANCEL PROVISAO CONTAS A PAGAR
1516	Campo 5: CT5_DC(C) :3
1517	Campo 6: CT5_DEBITO(C) :SA2->A2_CONTA

Importante: Neste exemplo o ALIAS SI5/CT5 estava posicionado em uma sequência do lançamento padrão que possui a variável apontada no evento de erro. Caso esta situação não ocorra, deverão ser avaliadas todas as sequências do lançamento através da consulta à respectiva tabela, e se caso necessário, avaliar o retorno dos "ExecBlocks" utilizados nos mesmos.

2º Passo: Avaliar o conteúdo do lançamento padrão que causou o erro

1. Avaliar se o conteúdo do campo está compatível com o tipo de preenchimento esperado pela ou definido para a rotina:

- ☐ Campo CT5_FILIAL: tipo caractere
- ☐ Forma de preenchimento: direto – sem a necessidade de ""
- ☐ Conteúdo informado: PF
- ☒ Conteúdo e forma de preenchimento compatíveis com o esperado pela rotina.

3º Passo: Interpretação da Análise e Próximas Ações

1. Pela análise do conteúdo do campo do ALIAS SI5/CT5 constatou-se o seguinte problema:

Causa do Problema: A rotina TRANSLCTA não está interpretando corretamente o conteúdo do campo CT5_FILIAL quando este possui apenas letras.

2. Como a “Causa do Problema” aponta o fonte como causador do erro, temos:

Solução / Próximos passos: Deverá ser avaliada a versão mais recente do fonte MATXFUNA.PRX, referente ao tratamento do campo CT5_FILIAL na função TRANSLCTA. Caso o fonte atual já possua o tratamento adequado, deverá ser aplicada uma patch do mesmo no RPO no qual ocorre a situação de erro, caso contrário deverá ser aberta uma solicitação de correção de erro para o fonte.

8. Informações Complementares

Na análise do arquivo error.log normalmente é necessário identificar algumas configurações do ambiente do cliente para poder determinar características que influenciam na ocorrência do erro.

Desta forma existem dois grupos de variáveis (Public e Private) utilizados pelo sistema que podem ser consultados para obter este tipo de informação.

8.1. Variáveis Publicas do Sistema

Variável	Tipo	Descrição	Exemplo de Conteúdo
__CLOGSIGA	Caractere	Conteúdo do parâmetro MV_LOGSIGA.	NNNNN
__CRDD	Caractere	Drivers de Banco de Dados utilizado pelo Protheus.	TOPCONN
__LOCALDRIVER	Caractere	Driver utilizado pelos arquivos locais / temporários do Protheus.	DBFCDX
__TTSINUSE	Lógico	Conteúdo referente ao parâmetro MV_TTS (S = .T. / N = .F.)	.T.
CARQTAB	Caractere	Listagem das tabelas abertas e o modo de compartilhamento das mesmas.	CTTE/ SE5E/ SD1C
CEMPANT	Caractere	Empresa ativa no sistema.	01
CFILANT	Caractere	Filial da empresa ativa no sistema.	05
CFOPENED	Caractere	Arquivos ativos no sistema.	RCESRNSI5SI1SI4SIC
CNIVEL	Numérico	Nível do usuário ativo no sistema.	5
CPAISLOC	Caractere	País para o qual o Protheus está configurado.	Brasil
CUSERNAME	Caractere	Usuário ativo no sistema.	Administrador
CVERSAO	Caractere	Versão do Protheus em uso.	AP5 5.08
DDATABASE	Data	Data com a qual o sistema está ativo.	2004/10/31
LAS400	Lógico	Se o Banco de Dados utilizado pelo Protheus é o AS400 em ambiente IBM.	.F.
MV_MOEDAx	Caractere	Conteúdo do parâmetro MV_MOEDAx.	REAIS
MV_PARxx	Múltiplos	Perguntas e respectivos conteúdos utilizados pela rotina.	Não se aplica
MV_SIMBx	Caractere	Conteúdo do parâmetro MV_SIMBx	R\$

8.2. Estrutura do Ambiente Protheus

O ambiente Protheus é composto basicamente pelos seguintes componentes:

❑ **BINÁRIO (BIN):**

Arquivos que compõe o núcleo do sistema Protheus, compostos pelos arquivos APxSVRxxx.EXE (principal) e APxRMT.EXE (dependente). A versão da BUILD do Protheus refere-se a estes arquivos.

❑ **RPO – Repositório de Objetos:**

Arquivo que contém os fontes compilados dos módulos que compõe o sistema Protheus e os fontes especiais conhecidos como LIBs. As LIBs são as bibliotecas de funções que em conjunto com o BIN definem as características técnicas e funcionalidades tecnológicas do sistema.

❑ **TOPCONNECT:**

Ferramenta de comunicação entre o sistema Protheus e os diversos bancos de dados relacionais com os quais o mesmo é implementado.

8.2.1. Componentes e Dependências

BINÁRIO (BIN)

- ❑ Define as versões de RPO compatíveis (datas de disponibilização);
- ❑ Define as versões de TOPCONNECT compatíveis (Builds do Topconnect).

RPO – Repositório de Objetos

- ❑ Depende da versão do BIN (BUILD) do Protheus em uso.

TOPCONNECT

- ❑ Depende da versão do BIN (BUILD) do Protheus em uso.

A versão do BIN (BUILD) do Protheus define quais datas de disponibilização de RPOs e quais versões (BUILDS) do TopConnect devem ser utilizadas.

8.2.2. Formas de Atualização dos Componentes

BINÁRIO (BIN)

- ☐ Disponibilização de nova BUILD (Server e Remote).

RPO – Repositório de Objetos

- ☐ Disponibilização de novo Repositório;
- ☐ Disponibilização de updates;
- ☐ Disponibilização de patches;
- ☐ Aplicação de patches emergenciais.

TOPCONNECT

- ☐ Disponibilização de nova BUILD.

9. Referências

9.1. Material Utilizado

- DEM – Documentação Eletrônica Microsiga
 - <http://dem.microsiga.com.br>
- Documentação de Atendimentos no PRC- ADM.FIN ACR.N1 e ACR.N2
- Manual Eletrônico – Advanced Protheus 7.10 e 8.11
 - Acesso interno: <http://helpintranet.microsiga.com.br>
 - Acesso através do site do ACR: <http://acr.microsiga.com.br>

9.2. Colaboradores

- Equipe do ACR.N2
- Equipe do ACR.N1
- Equipe do Desenvolvimento