Desenvolvimento (https://aprovador.com/blog/category/desenvolvimento)

Como utilizar todo o potencial do clientlog no Progress OpenEdge

clientlog · desenvolvimento · log · openedge · progress

WhatsApp LinkedIn Facebook Twitter E-mail

Todo mundo que trabalha ou trabalhou com Progress já passou pela necessidade de encontrar algum problema ou saber exatamente o que ocorreu em determinado ponto do programa.

O *debug* fica ainda mais complicado quando não temos acesso ao código fonte. Para estes casos, o Progress possui o clientlog, um recurso muito conhecido e utilizado. Sempre que precisamos de bate e pronto de ativar o clientlog, uma colinha é sempre bem-vinda.

O objetivo deste post é deixar uma consulta rápida de como ativar o clientlog no Progress Openedge e passar diversas dicas sobre como utiliza-lo de uma maneira mais prática.

Como funciona

O clientlog pode ser ativado inserindo parâmetros na PF ou em tempo de execução através do LogManager. Ele sempre vai funcionar da mesma maneira, independente da forma que for habilitado, o que muda é a sintaxe utilizada em cada caso.

Você pode utilizar o log de diversas formas e para diferentes objetivos. Eu separei um exemplo que vai atender a maioria dos casos, mesmo assim para situações específicas, consulte a documentação oficial.

Progress clientlog via PF

Habilitar o log pela PF é simples, você precisa apenas inserir os seguintes parâmetros no seu arquivo de conexão:

Ativar

- -debugalert
- -clientlog c:\tmp\clientlog.txt -clearlog
- -logginglevel 4
- -logentrytypes 4GLMessages,4GLTrace,DB.Connects,DynObjects.DB,DynObjects.XML,DynObjects.Other,DynObjects.CLASS,DynObjects.UI,File ID,ProEvents.UI.CHAR,ProEvents.UI.COMMAND,ProEvents.Other,SAX

A principal vantagem de se utilizar via PF é que quando a conexão é iniciada, todas as atividades já são gravadas no log.

A desvantagem é que você precisara alterar o arquivo .PF ou criar um arquivo exclusivo para o clientlog. **Atenção!** Cuidado com o tamanho do log pois ele pode crescer muito rapidamente! Falarei sobre isto mais adiante.

Progress clientlog em tempo de execução via LogManager

O Progress possui um recurso chamado LogManager que fornece uma série de métodos para trabalhar com logs. Usar o LogManager é muito simples, você precisa apenas abrir o editor (_edit.p) e executar o seguinte código:

Ativar

```
SESSION: DEBUG-ALERT = YES.

LOG-MANAGER: LOGFILE-NAME = "c:\tmp\clientlog.txt".

LOG-MANAGER: LOGGING-LEVEL = 4.

LOG-MANAGER: LOG-ENTRY-TYPES = "4GLMessages, 4GLTrace, DB. Connects, DynObjects. DB, DynObjects. XML, DynObjects. Other, DynObjects. CLASS, DynObjects. UI, FileID, ProEvents. UI. CHAR, ProEvents. UI. COMMAND, ProEvents. Other, SAX".
```

Desativar

```
SESSION: DEBUG-ALERT = NO.
LOG-MANAGER: CLOSE-LOG().
```

A principal vantagem é não precisar alterar nada no PF nem em qualquer outro arquivo de configuração. Você pode inclusive criar um programa que ativa ou desativa o clientlog. Outra grande vantagem é a facilidade em depurar apenas uma pequena parte da aplicação.

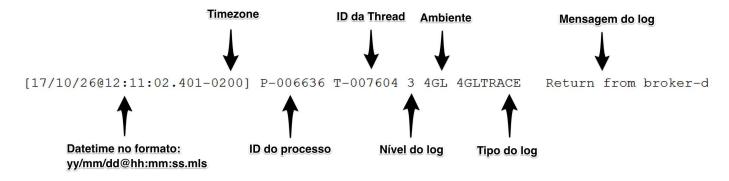
Detalhe de cada parametro

Para que serve cada um destes parâmetros dos exemplos acima?

| Via PF | Via LogManager | Para que serve? |
|--------------------|-----------------------------------|--|
| -debugalert | SESSION:DEBUG- ALERT | Caso o Progress retorne algum erro, toda a pilha de execução até o erro será exibida no log. |
| -logginglevel | LOG- MANAGER:LOGGING- LEVEL | Determina o nivel de detalhamento do log, onde: 0 - None - Não loga nada 1 - Errors - Somente erros Progress 2 - Basic - Somente log básico 3 - Verbose - Modo verbose dos valores definidos no logentrytypes 4 - Extended - Modo extendido dos valores definidos no logentrytypes |
| - logentrytypes | LOG-MANAGER:LOG- ENTRY-TYPES | Tipos de eventos que serão exibidos no log de acordo com o logginglevel definido. Um detalhe de todas as opções estão disponíveis no final deste post. |
| -clientlog | LOG- MANAGER:LOGFILE- NAME | Caminho e nome do arquivo do log que será gerado. |

Estrutura do Log

O log do Progres tem uma estrutura própria com data, cabeçalho e o conteúdo da mensagem. Veja o que significa cada parte desta estrutura:



Dicas

Algumas sacadas sobre o uso do clientlog.

Mudar o nível de detalhamento

Se você ativou o clientlog via PF, você ainda sim pode utilizar o LogManager. Neste caso o Progress deixará de manipular o log definido na PF e utilizará os parâmetros e definições do LogManager. Isto é muito útil para aumentar ou diminuir o detalhamento e para mudar o caminho e nome do arquivo de log.

Tamanho do arquivo

Como falei acima, é preciso tomar cuidado com o tamanho do log pois o arquivo cresce muito rápido, princialmente se você definiu o log para modo *extended*. Uma maneira de contornar isto é utilizar o método que sobreescreve o arquivo sempre que uma nova sessão for iniciada, pois, o padrão do Progress é nunca sobreescrever.

Via PF

Insira o parametro *-clearlog* após o nome do log. Exemplo:

-clientlog clientlog.txt -clearlog

• Em tempo de execução via LogManager

Após iniciado e definido o log pelo LogManager, execute o seguinte método:

LOG-MANAGER:CLEAR-LOG()

Escrever direto no log

Uma outra coisa legal é que você não precisa ficar dando *MESSAGE "passou aqui" VIEW-AS ALERT-BOX* no programa. Coloque as suas mensagens para saírem diretamente no clientlog. Para isto, após ativado o clientlog, execute o sequinte método:

LOG-MANAGER: WRITE-MESSAGE("passou aqui").

Especificação de cada função do clientlog

| Atributo | | Para que serve? | Nivel do log |
|-----------|----|---|-----------------------|
| 4GLMessag | es | Escreve no log todas os MESSAGE com VIEW-AS ALERT-BOX. | 2 - Basic |
| 4GLTrace | | Escreve no log as chamadas de RUN, PUBLISH, FUNCTIONS, and SUBSCRIBE e funções especificas. | 2 - Basic ou superior |

| Atributo | Para que serve? | Nivel do log |
|--|--|--|
| 4GLTrans | Escreve no log o quando uma transação começou, terminou ou foi desfeita. | 2 - Basic ou superior |
| DB.Connects | Escreve no log as aberturas e fechamentos de conexão com o bando de dados, incluindo o nome e id do usuário. | 2 - Basic ou superior |
| DynObjects.DB DynObjects.XML DynObjects.Other DynObjects.UI | Escreve no log a criação e remoção de qualquer tipo de objeto dinâmico. | 2 - Basic 3 - Verbose para detalhes da pilha |
| FileID | Escreve no log a abertura, fechamento e erros de programas e arquivos. | 2 - Basic e superior |
| ProEvents.UI.Char ProEvents.UI.Command ProEvents.Other | Escreve no log todos os eventos Progress de todas as categorias | 2 - Basic ou superior |
| SAX | Escreve no log as chamadas SAX. SAX é um parser de XML. | 2 - Basic |

Espero ter ajudado! Nos avisem caso tenha alguma observação, comentário ou correção!

Artigo por:

Marcelo Camargos in (https://www.linkedin.com/in/massreuy)

Aqui no Aprovador trabalho sempre para levar a melhor experiência de atendimento, segurança e simplicidade de produto para nossos clientes.

Vá muito além da aprovação!

Assine a nossa newsletter semanal e receba dicas e insights sobre negócio e tecnologia, além de todas postagens do blog.

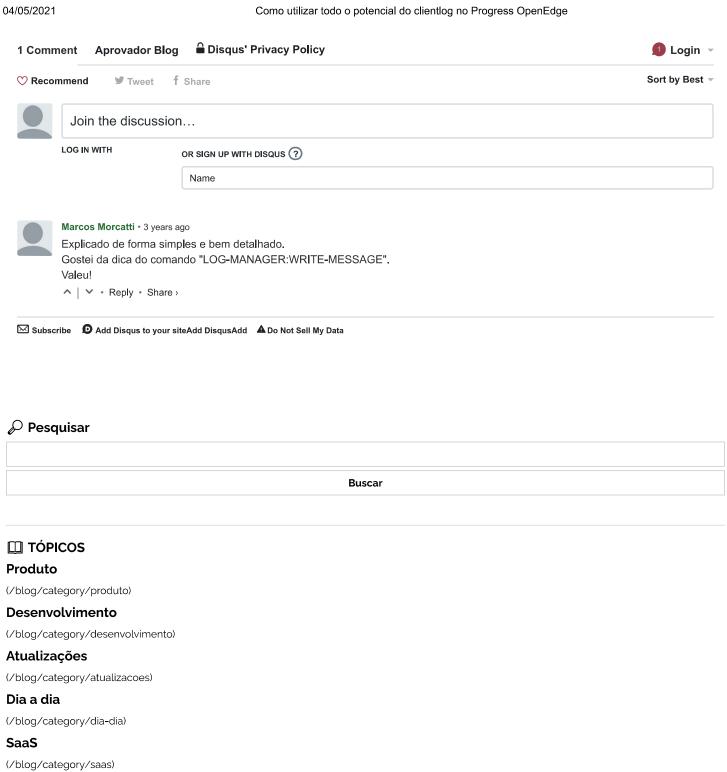
Nome

E-mail

Assinar Newsletter

Leia também:

- Driver ODBC Progress OpenEdge 32 bits em Windows 64 bits (driver-odbc-progress-openedge-32-bits-em-windows-64-bits?utm_source=blog&utm_medium=blog-post-related&utm_campaign=post-relacionado)
- Como configurar uma string de conexão Progress Openedge ODBC e JDBC (como-configuraruma-string-de-conexao-progress-openedge-odbc-e-jdbc? utm_source=blog&utm_medium=blog-post-related&utm_campaign=post-relacionado)
- Como acompanhar logs em tempo real no Windows (como-acompanhar-logs-em-tempo-real-no-windows?utm_source=blog&utm_medium=blog-post-related&utm_campaign=post-relacionado)



ERP

(/blog/category/erp)

Produtividade

(/blog/category/produtividade)

Totvs Datasul

(/blog/category/datasul)

Totvs RM

(/blog/category/totvs-rm)

Vagas

(/blog/category/vagas)

☐ SÉRIES

Guia de aprovações de compras no Totvs Datasul

(https://aprovador.com/blog/series/guia-de-aprovacoes-de-compras-no-totvs-datasul)

Esta é uma série especial de posts detalhando o funcionamento e parametrização das aprovações no processo de compras do Totvs Datasul.

| Receba novidades por e-mail | | | | |
|-----------------------------|--|--|--|--|
| Nome | | | | |
| E-mail | | | | |
| Assinar Newsletter | | | | |
| | | | | |



(https://aprovador.com/?

utm_source=blog&utm_medium=banner_lateral)

MAIS POPULARES

Como utilizar todo o potencial do clientlog no Progress OpenEdge.

(/blog/post/como-utilizar-todo-o-potencial-do-clientlog-progress-openedge) 24/11/2017

O Aprovador agora fala Inglês!.

(/blog/post/o-aprovador-agora-fala-ingles) 13/11/2017

04/05/2021

Talvez você não precise dessa customização.

(/blog/post/talvez-voce-nao-precise-dessa-customizacao) 01/10/2017

ROI e métricas intangíveis em projetos de TI

(/blog/post/roi-e-metricas-intangiveis-em-projetos-de-ti) 27/05/2016

APROVADOR.COM (HTTP://APROVADOR.COM)

Home (/)

ERPs (/integracoes)

Clientes (/clientes)

Sobre o Aprovador (/sobre)

Entre em contato (/fale-conosco) Teste grátis (/teste-gratis)

APLICATIVOS



(/link/app-ios)



(/link/app-android)

CONTATO

**** 31 3144-7205

(https://www.linkedin.com/company/aprovador)
(https://www.facebook.com/aprovador)
(https://www.youtube.com/channel/UCuXbR7p6GtL6SCWdFxTqGjQ)

Datasul, RM e TOTVS são produtos e marcas registradas de propriedade da TOTVS S.A. Logotipos TOTVS, Datasul e RM são de propriedade da TOTVS S.A.