

**BadLink**

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> BadLink		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 22, 2024	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>BadLink</b>	<b>1</b>
1.1	Manual - main menu . . . . .	1
1.2	Prawa autorskie i rozpowszechnianie . . . . .	1
1.3	Do czego ten program służy? . . . . .	2
1.4	Opis programu . . . . .	3
1.5	Wymagania, biblioteki, dane techniczne . . . . .	4
1.6	Komunikaty pîentuf . . . . .	4
1.7	Co w przyszłości? . . . . .	5
1.8	Historia programu . . . . .	5
1.9	Co to jest BlaBla? . . . . .	5
1.10	Pozostałe informacje . . . . .	6
1.11	DiskSalv . . . . .	6
1.12	QuarterbackTools . . . . .	6
1.13	Walidacja . . . . .	7
1.14	Nagîówek . . . . .	7
1.15	Rigid Disk Block . . . . .	7

## BadLink

1. ~Prawa~autorskie~i~rozpowszechnianie.~::~
2. ~Do~czego~ten~program~służy?~::~
3. ~Opis~programu.~::~
4. ~Wymagania,~biblioteki,~dane~techniczne.~
5. ~Komunikaty~płentuf.~::~
6. ~Co~w~przyszłości?~::~
7. ~Historia~programu.~::~
8. ~Co~to~jest~BlaBla?~::~
9. ~Pozostałe~informacje.~::~

## 1.2 Prawa autorskie i rozpowszechnianie

Ten program podlega Standard~Amiga~FD~Software~Copyright~Note.

Ten program jest Freeware, jak zostało to określone w punkcie 4a.

Jeśli szukasz więcej szczegółów, proszę przeczytać AFD-COPYRIGHT (wersję 1 lub nowszą).

```
~Prawa~autorskie~  
~Dystrybucja~~~~~  
~Oświadczenie~~~~  
~Opłaty~~~~~
```

### 1.3 Do czego ten program służy?

System operacyjny Amigi jest na pewno jednym z najlepszych systemów operacyjnych. Jest jednak (nawet w najnowszej wersji 3.1) dość stary i ma trochę niedociągnięć. Przykładem jest obsługa urządzeń zewnętrznych (zwłaszcza nośników danych). KS bardzo słabo radzi sobie z błędami na dyskach (zarówno programowymi, jak i sprzętowymi). Co prawda istnieją programy, które mają ogromne możliwości w tym temacie (DiskSalv, QBTools), ale również one nie radzą sobie z jedną, bardzo ważną rzeczą.

Wiadomo, że wcześniejsze RESETY podczas zapisywania danych na nośnik powoduje utracenie jego walidacji. Jest to bardzo denerwujące, zwłaszcza w przypadku dużego twardego dysku. Trzeba przyznać, że jest to lepsze rozwiązanie niż to znane z PC (ostatnio mój kolega stracił wszystkie dane na wszystkich partycjach), ale denerwujące. Mało tego – często dysk nie chce nam się w żaden sposób zwalidować, bo występują różne błędy logiczne, np. "Block nnn used twice", "Block nnn checksum error", czy "Block nnn out of range". O ile w 2 pierwszych przypadkach 2 wymienione wyżej programy radzą sobie wyśmienicie, to komunikat trzeci jest złą, której nie można zwalczyć żadnym znanym mi dotychczas programem. O co więc chodzi?

Wiadomo, że przy nagłym przerwaniu dostępu prądu przy zapisie, twardy przyjaciel może trochę oszaleć i w ostatniej chwili zapisać trochę bzdurek. Pół biedy, jeśli trafi na blok z danymi, najczęściej jednak trafia do tzw. headeru, czyli bloku z informacjami o pliku/katalogu. Co się wtedy dzieje. Plik staje się bezużyteczny, gdy posiada odwołania do złych bloków. Zamiast do bloków dobrych, odwołuje się do powstałej sieczki (np. \$FFFFFFFE, \$1 itp.). No tak, ale co to ma wspólnego z walidacją? Ano ma. Walidator jest na tyle zmyślnym programem, że sprawdza we wszystkich "headerach" linki i ocenia czy są one dobre (czy mieszczą się w przedziale bloków, jakie ma partycja). Niestety, na tym jego "inteligencja" się kończy. Jeśli natrafi na fałszywe odwołanie, po prostu wypisuje komunikat o błędzie ("Block nnn out of range") i pozostawia partycję w stanie niezwalidowanym (tzw. read only). A co robi użytkownik? Skoro nie pomagają żadne programy zaczyna się denerwować i planować sformatowanie partycji połączone z jej wcześniejszym backupem na dyskietki. Ile to roboty, zwłaszcza przy dużych pojemnościach, chyba nie muszę wspominać. A wystarczy jeden, mały program (Bad Link) i jeden duży program (QuarterbackTools). Dzięki nim udało mi się w ciągu dwóch dni (wraz z programowaniem BL i nie licząc wcześniejszych 3 tygodni nie dotykania się do komputera) zwalidować partycję i odzyskać wszystkie dane (no, prawie, ale to wina AmibackTools).

Jak działa Bad Link? Na wskazanej partycji stara się znaleźć wszystkie bloki z nagłówkami (zarówno drawer, extension, jak i fileheader) i sprawdza, czy nie występuje w nich odwołanie do błędnego bloku (podanego przez użytkownika). Jeśli tak, do od razu daje o tym znać. Wiedząc już, jaki blok powoduje ten galimatias wkraczamy do QBTools. W opcji EDIT na odpowiedniej partycji skaczemy do wskazanego przez Bad Linka bloku i zamieniamy wszystkie odwołania do błędnego bloku na zera (UWAGA! Należy pamiętać że zmienić możemy jedynie pola "File Key", "Ext Key", "First block key" (to dla fileheader i extension), "Hash table key" (tylko dla drawer). Jeśli znasz się na rzeczy, możesz ustawić na odpowiedni argument "Block count" (nie jest to chyba konieczne, walidator powinien sobie z tym poradzić) Teraz partycja powinna się spokojnie zwalidować.

UWAGA! Jeśli jesteś początkujący w tym temacie to lepiej nie dotykaj się do QBTools! Nieumiejętnie wpisując liczby możesz zniszczyć niektóre dane. Poczekać na nową wersję Bad Linka, która wszystko zrobi za Ciebie automatycznie!

## 1.4 Opis programu

Program uruchamiany jest wyłącznie z poziomu Shella. Oto jego opcje:

LINK/A/N,SCSINUM=SC/K/N,PARTITION=PA/K/N,PROGRESS=PR/S

Gdzie:

LINK - argument konieczny, jest linkiem, jakiego ma szukać program w blokach z headerami. Wpisz tu po prostu numer bloku, jaki podaje Ci walidator (w komunikacie "Block >nnn< out of range"). Często będzie to po prostu 1 (jedynek). Przykład: 1233340

SCSINUM - Tu podajesz numer urządzenia SCSI. Przy jednym dysku jest to oczywiście 0 (wartość domyślna). Przykład: SC=1

PARTITION - Numer partycji. Domyślnie jest to 1. UWAGA! Program szuka partycji wykorzystując funkcję tzw. "low-level"! Znaczy to, że nawet jeśli wyliczyłeś jaką partycję w "Boot menu" to program może ją znaleźć i przeszukać. Dzięki temu nie masz problemu z ciągłą walidacją zajmującą twardego. Jeszcze jedna ważna rzecz. Musisz podać numer partycji, ale symbolizujący jej FIZYCZNE położenie, a nie numer w nazwie. Jeśli np. masz partycje DH0:, DH1:, DH2: to chcąc przeszukać DH2 musisz wpisać dla tego argumentu 3, a dla DH0 - 1. Przykład: PA=2

PROGRESS - Normalnie program działa nie podając informacji o zaawansowaniu w pracy (spowodowane jest to tym, że użytkownik może chcieć kierować strumieniem do pliku, co spowodowałoby jego szybkie "zapchanie"). W celu "włączenia" informacji o zaawansowaniu (podany będzie aktualnie przeszukiwany blok) po prostu wpisz nazwę tego switcha. Przykład: PR

Uwagi:

Jeśli w czymś się pomyliłeś, lub uważasz że coś jest nie tak, to nie panikuj! Po prostu wciśnij na raz "CTRL+C". Program automatycznie zakończy pracę.

Jeśli program znalazł zły blok, to nie wstrzymuj jego pracy! Bardzo możliwe że istnieją kolejne bloki ze złymi danymi! W moim przypadku były dwa takie bloki!

Na początku program wypisuje kilka przydatnych informacji o partycji, jak jej nazwa, ilość bloków, blok w RDB zawierający informacje o niej.

## 1.5 Wymagania, biblioteki, dane techniczne

Do uruchomienia programu niezbędny jest system 2.0+ i trochę wolnej pamięci (ok. 50KB). Program nie wymaga żadnych zewnętrznych bibliotek i urządzeń. Korzysta tylko ze "scsi.device".

Program testowany był na komputerze: Amiga 1200, KS3.0, karta Blizzard III, 8MB Fast, HDD Conner 1080MB (1.30).

Szybkość programu nie jest zbyt wielka, ale chyba wystarczająca. Na moim komputerze z dość wolnym HD Connera (1.2MB/sek.) 20000 bloków przeszukuje się w ok. 18 sekund. Cała partycja 0.5GB zapełniona w 98% przeszukiwała się ok. 15 min. (m.w. tyle co walidacja)

Program napisany został w języku programowania Amiga E 3.2a. Źródówka zajmuje 6632 bajty, na które składają się 330 linie kodu (z komentarzami).

## 1.6 Komunikaty piontuf

Cannot open IO port - prawdopodobnie nie masz wogóle pamięci :)

Cannot open IOREquest - Nie da się otworzyć IOReq. Patrz wyżej.

Cannot open device - Albo nie masz twardziela, albo podałeś złý numer SCSI

Cannot read from device - Nie da rady wyczytać danych z twardego. Czyby bady? A może coś jest złe w RDB?

Cannot find RDB! - Twój HDk nie ma Rigid Disk Blocku. Módl się, żebyś nie stracił wszystkiego co masz. ZAWSZE rób kopię RDB na dyskietki!

This partition does not exist - Podałeś numer partycji, która nie istnieje. Jeśli nie jesteś pewien kolejności partycji, sprawdź to HDToolsBoxem.

No memory for buffer! - Program nie może zaalokować 512 bajtów RAMu. Oj chłopcze, spraw sobie jakieś rozszerzenie!

Unknown error: nnn, sss - Błąd o którym nie mam pojęcia. NATYCHMIAST opisz mi całą sytuację i podaj wyświetlone przez program informacje!

## 1.7 Co w przyszłości?

Program ten powstał w 3 dni tylko i wyłącznie z potrzeby (padł mi HDk) i ze słyszenia, że wiele ludzi miało podobne problemy i nie umiało sobie z nimi poradzić (mi nie pomogło nawet grono ludzi z MA! :) Chcę tylko powiedzieć że jest to wersja całkowicie przeznaczona dla ludzi, którzy orientują się trochę w systemie i wiedzą chociaż co to jest link (czy inaczej "file key"). W przyszłości, jak mi na to czas pozwoli poprawię program tak, by działał dużo szybciej, sprawniej, lepiej, miał więcej możliwości, a w szczególności, by sam potrafił naprawiać bloki, bez potrzeby ręcznego używania QBTools. Nie przewiduję dla programu GUI bo nie jest to chyba najważniejsze. Czekam także na Wasze opinie, do których z pewnością się ustosunkuję!

## 1.8 Historia programu

wersja 1.0                      08.12.1996                      o Pierwsza wersja publiczna.

## 1.9 Co to jest BlaBla?

BlaBla jest to grupa zrzeszająca programistów piszących programy współpracujące z systemem. Gwarantuje to poprawną pracę na wielu modelach Amig, oraz zapewnia zachowanie kompatybilności "w górę". Pozwala to również na korzystanie z tak ważnej cechy, jaką posiada Amiga, jak multitasking. Nasze programy mają zazwyczaj status PublicDomain, FreeWare lub ShareWare, co pozwala na ich darmowe rozpowszechnianie.

Grupa posiada własne konto Internetowe i BBS. Nasze najnowsze programy można również znaleźć w wydawanym przez nas pakiecie polskich programów użytkowych PolWare. Zapraszamy również do lektury redagowanego przez nas magazynu dyskowego Izviestia. Traktuje on głównie o programowaniu i zawiera opisy wielu ciekawych programów.

Zainteresowanych współpracą z naszą grupą, bądź chętnych do wymiany doświadczeń i uwag na temat programów, prosimy o kontakt z którymś z członków BlaBla.

W skład grupy wchodzi (X'96):

Apacz/Fire & BlaBla	(Michał Kopacz)
Kordi/DuckRed & BlaBla	(Kordian Adamczyk)
Kysy/AM & 2xBla	(Krzysztof Habowski)
LeMUr/Fire & bla\$^2\$	(Łukasz Prokulski)
Novi/True Genius & Phx & S2B	(Krzysztof Nowak)
Reese/SubBlaBla	(Karol Bryd)
Scott/Inferno & BlaBla	(Marcin Ochocki)
Thufor/BlahBlah	(Dariusz J.Garbowski)
Tomash/Art-B & BBla	(Tomasz Korolczuk)
Warhawk/SubBlaBla	(Przemysław Jeń)



## 1.10 Pozostałe informacje

Chciałbym bardzo podziękować wszystkim przyjaciołom, a w szczególności Chomikowi, który objaśnił mi całą sytuację z walidacją i poinformował, że ręcznie usuwał ten błąd przez dwa tygodnie (przy duuuużo mniej zapchanym twardziewie). To On zmobilizował mnie do pracy! Chciałbym też podziękować Słoniowi za pomoc w reklamie Polware oraz całemu ACS za to że nie są tak bezwzględnie komercyjni jak MA. Niewielkie ;) podziękowania należą się też Romanowi Sadowskiemu za chęć ;) pomocy. Pozatym pozdrowienia dla całej Blabli!

Jeśli chcesz się ze mną skontaktować w wiadomym celu (informacje o błędach, chęć poznania mnie (:), wstąpienia do BLABLA itp.) to pisz na ten adres:

Tomasz Korolczuk  
ul. Wyspiańskiego 4/13  
08-300 Sokołów Podlaski

Możesz też przesłać jakieś wiadomości internetem. Poniżej umieszczam 3 adresy moich kolegów z BLABLA. Zaznaczcie że to informacja dla mnie :) :

Kordi	kordi@lodz2.p.lodz.pl
Thufor	thufor@zeus.gliwice.edu.pl
Warhawk	jez@szmaragd.zdz.katowice.pl

Dzięki za wszystko!

## 1.11 DiskSalv

Program ten napisany został przez jednego z konstruktorów Amigi - Dave'a Hayniego. Służy on do naprawiania utraconych danych, "oczyszczania" twardziela i wielu innych rzeczy. Niestety, nawet DS w najnowszej wersji (IV) nie radzi sobie z błędem walidacji, który wystąpił w moim przypadku

## 1.12 QuarterbackTools

Quarterbacktools jest konkurencją programu DiskSalv. Niestety, od roku 1993 nie ukazała się już żadna nowa wersja programu. Mimo to QBTools jest pewnym, przyjemnym programem, któremu na pewno można zaufać (w tym przypadku DiskSalv jest wiele bardziej niebezpieczny). Podobnie jak DiskSalv nie sprawdza poprawności linków w plikach.

QBT ma jedną bardzo przydatną opcję - edycji partycji. Można w bardzo łatwy sposób zmieniać zawartość dowolnego dysku (np. dobierać się do nagłówków plików, ich danych). Dzięki tej opcji możliwa staje się naprawa twardego przyjaciela z pomocą BadLinka.

## 1.13 Walidacja

Inaczej - "poprawianie". System Amigi posiada wielkie zabezpieczenie przed utratą danych w sytuacji, gdy nagle "wysiadzie" zasilanie. Gdy system zapisuje coś na dysku, ustawia jego "stan" na "roboczy" - ustawia bit walidacji na NIE. Po co to? Po to, że gdy nagle zapisywanie zostanie przerwane (reset itp.) to system, będzie wiedział, że na dysku jest coś nie tak. Do akcji wkracza wtedy "walidator", który przeszukuje twardego dysku w poszukiwaniu błędów, wyklucza je i ustawia bit walidacji na TAK. W teorii pięknie to wygląda, jednak nie powstałby mój program, gdyby wszystko sprawdzało się w praktyce. Walidator jest bardzo ograniczony (także ze względu na brak miejsca w ROMie) i nie radzi sobie zbyt dobrze z powierzonym zadaniem.

## 1.14 Nagłówek

Každy plik, katalog na dysku ma zapisane w pewnym miejscu dane, które mówią o jego położeniu fizycznym na owym dysku. Są tam też zawarte informacje o dacie jego weryfikacji, flagach protekcji, komentarzu itp. Najważniejsze są jednak dane o blokach, jakie wykorzystywane są przez plik (tzw. linki). To dzięki tym danym komputer wie, co, skąd ma czytać. Właśnie te zapisane linki są często powodem złego działania walidacji.

Linki mają różne typy. Używając QBTools w sposób opisany w tej instrukcji należy pamiętać, że zmieniać można TYLKO podane przeze mnie (każde pole w bloku jest przez QBTools dokładnie opisane). Niepoprawnie wpisane dane mogą być przyczyną kolejnych strat. Aby uzyskać dokładne informacje na temat linków, sięgnij po specjalistyczną literaturę (to za długi temat aby przedstawiać go w tej instrukcji).

## 1.15 Rigid Disk Block

Rigid Disk block jest to najważniejsza część całego twardego dysku, normalnie niedostępna dla użytkownika. Zawarte są tam przeróżne informacje, począwszy od mniej ważnych (data stworzenia, nazwa producenta), a skończywszy na tych najważniejszych (położenie partycji, parametry twardego dysku). Wiele wirusów próbuje zniszczyć RDB. Musisz wiedzieć, że jego utrata powoduje bezpowrotne zniszczenie wszystkich danych i zmusi Ciebie do ponownego "instalowania" twardego dysku HDToolsBoxem! Pamiętaj o tym, aby zawsze trzymać kopię RDB na dyskietce. Możesz to zrobić np. moim programem o nazwie RDBArc.