DELTA software technology

NEWSLETTER
ISSUE JULY 2026

IN THIS ISSUE

Get in touch with us

Delta Software Technology GmbH
Eichenweg 16
57392 Schmallenberg
Germany

phone +49 2972 9719-0
fax +49 2972 9719-60
e-mail info@delta-software.com

delta-software.com

More newsletters and our newsletter administration can be found here:

delta-software.com/newsletter

1 50 Years of Delta ADS: Supporting the Developer

Delta ADS turns 50

Delta ADS - Review

This year, 2026, we celebrate 50 years of Delta ADS. In this years newsletters we will look back on the history of Delta ADS and reflect on its features that have contributed to its long lasting success.

Half a century ago, programming a computer was a special kind of magic for most people. It also was an individual and very creative process with pen, paper and punch cards. Getting complex programs functioning correctly was a difficult and time-consuming process and came with considerable costs that did not allow for the compiler to be used for error finding.

The use of terminals to write programs was a big improvement in the development process. It provided a better overview of the program logic without having to compile and print a listing first. This overview was limited of course by the maximum of 24 lines a terminal was able to show at the same time. This was not sufficient for complex decisions which gave rise to the first tools to support the programming. Although such tools were available for many programming languages, the main field for these tools was in the area of the most widely used programming language for business applications: COBOL.

1971: The First Idea

In 1971, Dr Reinhold Thurner of Zürich University published[1] about the use of decision tables in the software development which led to DETAB/GT to generate the programming code based on a more abstract definition language . By 1976 DETAB/GT reached 100 installations, the same amount of installations was achieved by its competitor VORELLE[2].

Generating a complex decision structure for COBOL programs was a great help, especially when keeping in mind that in these days COBOL statements were terminated by a single point and complex decisions could easily run over 24 lines of code. It also showed that effective programming needed more. A major step was not in the use of tools but in the way, programs were structured. In an environment dominated by card readers, tapes and report

writers (hard disks were expensive and had only limited capacity), each program depended on the structure of the input and output data. The fact that each programmer used his or her personal style and creativity did not make the maintenance of these programs easy. Some kind of craftmanship and best practices was needed. This led to a number of Structured Programming initiatives. The most widely used was Jackson Structured Programming (JSP)[3] that Michael Jackson published in 1975.

At the same time Dr. Thurner launched a generator system that, based on the principles of structured programming, was able to produce much of the needed code. It used an abstract description of the structure of the data, in a language specifically defined for that task. The product was called DELTA[4]. It solved the problem that Dijkstra addressed in 1968. The programmer did not need to program and maintain any GO TO statement, only the generator produced them for the executable program[5].

Over the years, Delta (later called Delta/ADS) was extended with a number of standard generators for standard functionality in Business Programming. PSD (Program Schema Description) and GRU (Control Break Processing) helped programmers to keep an overview of their programs and to focus on the unique functionality they have to implement while not being bothered by mechanisms needed to implement it.

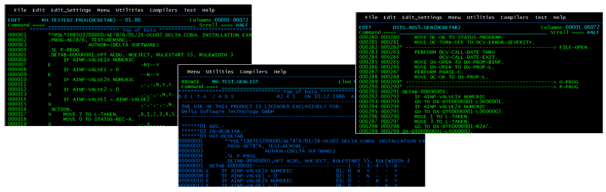
Platform independence - Thanks to Generators

A major step in the development of Delta ADS was the concept of platform independence by the use of generators and macros. Back in the old days, each mainframe (compiler) had its own deviations of the COBOL standard. A program written for platform A was thus not usable on platform B without changing the source code. Not a big deal if you develop your software inhouse for your own use but it is for a software supplier that wants to install his software on many platforms. The use of the generators and macros in Delta ADS to deal with the platform specific aspects of the implementation meant that a program for many different target platforms could be maintained in one single source code object.

Platform independence did not only benefit software suppliers but after some time many Delta ADS users discovered that this possibility also facilitated a platform migration. The fact that an existing application could be migrated easily to a new platform when using Delta ADS is an important reason why many installations are still actively used, some for over 40 years, and that Delta/ADS is still alive and kicking after 50 years.

> "I am pleased that after 50 years, Delta ADS is still being used successfully for the development and migration of large and complex applications."
>
> **Dr. Reinhold Thurner**

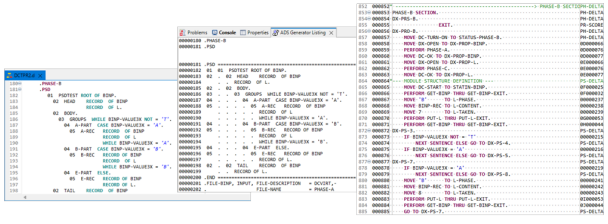
The Future - and beyond



DETAB in ISPF

Like Delta DETAB, Delta ADS for COBOL and PL/I remains a part of our product portfolio. Over the years an estimated 2+ billion unique lines of code have been generated with our generator system, a number that is still growing each year. Initially, DETAB and Delta ADS were only available for the platforms for which they generated code. Today they are also integrated into modern development environments such as Eclipse or Visual Studio and can generate cross-platform.

As the advantages of this generative approach to software development is still valued by our customers we remain committed to supporting them with our tools. The current release 6 of Delta ADS will certainly not be the last as we will continue to find solutions for the challenges our customers experience in their software development process.

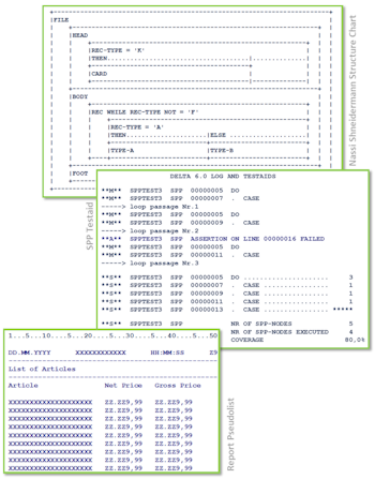


Delta ADS in Eclipse

NEWSLETTER
ISSUE JULY 2026

50 Years of Delta ADS: Supporting the Developer

This year, 2026, we celebrate 50 years of Delta ADS. In our first retrospective we focussed on the techniques that played a big role in the area of code produce. Now we will look into another important part of Delta ADS, besides producing code: supporting the developer with tooling, prototyping, documentation and testing.



The first domain that Delta ADS took care of is the production of documentation. During the generation process, a number of different sorts of documents can be produced. Depending on the Delta ADS processors used, e.g. a program flow chart is created, an overview of all available locations and their usage becomes available and, last but not least, SPP has the option of generation a Nassi-Shneiderman diagram of a SPP-routine.

All these features enabled the developer to check the actual program against the design he made. But being able to communicate with the end-user about the results of a program has always been an important but difficult issue. Results could only be shown after running the program which needed a functional program and available data. Adapting afterwards was a tedious and error prone process. Therefore, Delta ADS allows for prototyping of reports and screens which produce the simulated results during the program generation. This gives the feedback at a much earlier stage, reducing the effort to correct the implementation.

Correcting existing programs has always been an important, time consuming but undervalued part of a programmer's job.

Delta's founding father Dr. Reinhold Thurner already found out in an early stage that: 'for the correction of an error, 20% of the effort is needed to determine that there is an error and to pass the issue to the correct people. Next, in terms of time consuming, is to find the component that causes the error. The effort needed to fix the error is very often limited (but as we are in a hurry we tend to introduce new errors)'.

In a time where interactive debuggers were not available, finding an error was often a case of inserting display statements in the code to see where the program was when the error occurred. After fixing the flaw, these display statements had to be removed again. When another error came up, this process had to be repeated again.

Using a generator for the code produce opened up a solution for this time-consuming process. By activating one of the TEST-options (for the complete program or just one SPP-routine), the program will protocol each passed SPP-construct. This gives the programmer the much-needed insight in what a program did at runtime (and much faster compared to debugging) without having to program display-statements for that.

Additionally it can also produce statistics showing what logic was executed during the test run, giving an idea about the validity of the test and about general test coverage.

From the start, Delta ADS aimed at supporting the programmer to produce structured, well documented and maintainable code. A mission that we, after 50 years, remain to continue.

History

Article from Computerwoche, January 10, 1975 - Delta Author Dr. Reinhold Thurner:

„Multifunktionales Generator-System für moderne Software-Technologie: Ein Superding für die Avantgarde“

50 Years Delta ADS

Life starts at 40 something: Delta ADS turns 50 this year

Developing software is looking to the future, but sometimes it is good to turn around and look back. As we did, we discovered that the foundations of our products were made almost half a century ago.

Read more

Download

Sources

[1] Thurner R., „Entscheidungs-Tabellen: Aufbau-Anwendung-Programmierung“, VDI- Taschenbuch T33, VDI Verlag, Düsseldorf, 1971.

[2] Jeweils 100mal installiert: Vorelle und Detab/GT | Computerwoche, 02.07.1976

[3] Jackson, MA (1975), Principles of Program Design, Academic.

[4] Multifunktionales Generator-System für moderne Software-Technologie: Ein Superding für die Avantgarde | Computerwoche, 10.10.1975

[5] Dijkstra, Edsger W. (March 1968). "Letters to the editor: Go to statement considered harmful" (PDF). Communications of the ACM. 11 (3): 147–148. doi:10.1145/362929.362947. ISSN 0001-0782. S2CID 17469809.

p. 147, *"The unbridled use of the go to statement has as an immediate consequence that it becomes terribly hard to find a meaningful set of coordinates in which to describe the process progress. ... The go to statement as it stands is just too primitive, it is too much an invitation to make a mess of one's program."*

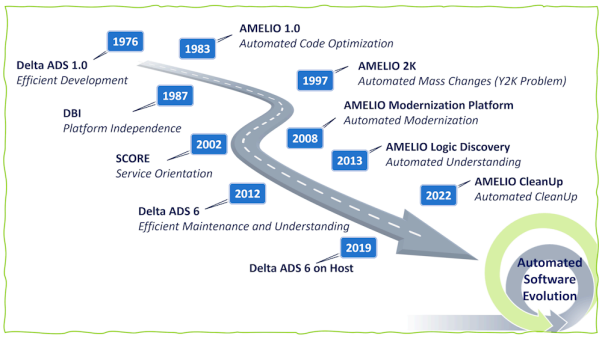
2 From Software+Methode to Delta Software Technology in 40 Years

In 1985, Winfried Buhl and my parents, Eva and Rüdiger Schilling, registered the company Software+Methode in the commercial register. The name was intended to reflect the mission; from the very beginning, the aim was to make software development more efficient through a methodical approach. With plenty of good ideas and a great deal of enthusiasm, they began their work on March 1, 1986 - that is 40 years ago.

Rüdiger & Eva Schilling and Winfried Buhl [2001]

While Delta was not our very first customer, it was one of the first - and an important one at that. As early as 1987, Software+Methode supplied DBI as a component for Delta’s generator system, ADS. In 1995, we acquired Delta from its then-parent company. This acquisition, combined with further strategic restructuring, led to the formation of Delta Software Technology in 2000 - the company as it exists today.

With the acquisition of Delta, ADS also became part of our portfolio. Delta ADS remains in use by many customers to this day and is continuously being further developed and maintened by us. The development of AMELIO also began in the 1980s. Initially intended for code optimization, it evolved into a tool for automated modernization during the 1990s and 2000s, and today serves as our key to automated software evolution.



50 Years Abtraction & Automation

NEWSLETTER
ISSUE JULY 2026

Constants in Flux

Software development is known for its fast pace. New technologies, processes, and methodologies enter the market in rapid cycles, and companies come and go. We, too, have repeatedly adapted and kept pace with the times. Yet two constants have remained from the very beginning:

- To this day, abstraction and automation are our keys to efficient, high-quality software development and evolution.
- To this day, the company remains 100% owned by the founding families Buhl/von Cieminski and Schilling - and is managed by me, Dr. Daniela Schilling, representing the second generation of leadership.

Rüdiger Schilling, Winfried Buhl & Daniela Schilling [2013]

And in the Future?

A look into the future is like the famous gaze into the crystal ball!

Looking ahead, we will continue to maintain, optimize, and enhance our proven tools and solutions - particularly ADS and AMELIO and complete new technologies to create further possibilities. In everything we do, it is you - with your wishes and requirements - who drive and motivate us and help shape our direction. And we intend for that to remain the case in the future.

We look forward to many more years of trusting and constructive cooperation with you.

Warm regards,

Daniela Schilling

3 Delta ADS Webinar - Refresh & Connect

Review of the Webinar held on June 17, 2026:
50 Years Delta ADS

"Longevity" is a buzzword that refers to a long life - and, related with that, a healthy life.

It's not often that this term is associated with software. But there is one application that's turning half a century old: Delta ADS.

One might be quick to assume that it is completely outdated. With ADS, that is definitely not the case. Since its initial release, ADS has undergone countless innovations and updates. In its current version, 6.6.9, it meets all the requirements of today's technologies.

At the start of the webinar, our CEO, Dr. Daniela Schilling, provided a comprehensive overview and highlighted key milestones such as platform independence and improvements driven by customer needs. She explained enhancements in ADS 6 that improve the code's readability, maintainability and scalability, including a macro source optimizer and the removal of parameter limits.

At a glance:

- ADS 6 is downward compatible with version 4.x and focuses on improving code readability.
- The new macro syntax enables named parameters and free-form formatting with indentation.
- A macro source optimizer automates the code formatting to improve maintainability.
- Parameter limits have been significantly increased; for example, the number of local parameters per macro has been raised to 2,500.
- The nesting depth for macros has been increased from 10 to 50, and the number of code lines per module from 400 to 1,000.
- The code generation now offers options to reduce GOTO statements and improve readability.

- The optimizations include the generation of IF and SELECT CASE statements using newer syntax standards.
- Customers can selectively enable these options to avoid a mandatory full regeneration.
- The improved code is shorter and easier to read, which helps developers with maintenance.
- This represents a paradigm change in the handling of generated code in ADS.

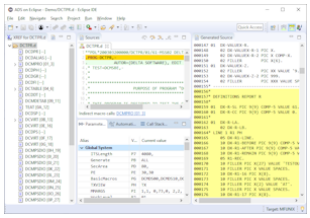
Integration with Modern Development Environments

To bridge the gap between ADS and modern development workflows, the team has developed ADS plugins - including plugins for Eclipse and Visual Studio - named Delta IDS. These plugins offer intelligent code editors, integrated generation processes, and debugging tools that enable developers to work in familiar environments. The integration supports features such as error highlighting, direct navigation to code issues and post-generation analysis to trace the origins of the code.

- ADS plugins are available for Eclipse and Visual Studio and support modern IDE features.
- Developers can generate code, view errors, and navigate directly to problematic lines within the IDE.
- Post-generation debugging tools allow you to trace generated code back to source macros and parameters.
- The integration supports the call of sources via FTP or other URLs.
- These tools improve the developer productivity and reduce context change.

Live Demonstration of ADS on Eclipse

Our chief architect, Mr. Martin Herbst, presented a live demo showcasing the ADS plugin for Eclipse. He demonstrated project navigation, code editing



with syntax highlighting and folding, error detection and the impact of the macro source optimizer on code readability. He also highlighted the post-generation debugger, which records macro performance and enables detailed inspection of the code generation flow, parameter values and call stacks. This provides an efficient debugging and maintenance.

- Demo of the Project Explorer and Editor features of the ADS Eclipse plugin.
- He demonstrated error lists and direct navigation to source code errors.
- The macro source optimizer improves code indentation and parameter naming for better readability.
- The post-generation debugger records macro performance and highlights executed code paths.
- Users can navigate call stacks and inspect parameter values to understand code generation.

Future Developments and Customer Engagement

Daniela Schilling outlined current and future development plans, including support for new COBOL dialects such as P3/COBOL and modern databases such as PostgreSQL. The team examines object-oriented goals beyond COBOL and PL/I. The focus is on customer-driven development, which encourages users to communicate their needs and challenges to guide future improvements. A release date for Version 7 has not yet been set.

- New goals include support for P3/COBOL and PostgreSQL databases.
- Work is underway to enable object-oriented code generation beyond traditional languages.
- Development is heavily customer-driven, with feedback determining priorities.
- Version 7 has not yet been planned or scheduled.
- The team invites customers to share their requirements and issues so we can work together to find solutions.

NEWSLETTER
ISSUE JULY 2026

The team emphasizes ongoing development efforts, including support for new COBOL dialects and databases, as well as plans for object-oriented code generation—which were all driven by customer feedback.

The pipeline for even more functionality is in place, and we warmly invite you to share your requirements with us. We are looking forward to the next 50 years!

You were unable to attend?

No problem. If you are interested in getting the presented information, please contact us directly.

Some Success Stories:

YOUPLUS Unifies its Development Environment and Updates Delta ADS

A modern development environment was the goal for the software developers of YOUPLUS Assurance Schweiz AG. Together with Delta Software Technology, the previous development environment was replaced by ADS on Eclipse and at the same time the generator system Delta ADS for COBOL was updated.

More here:

https://delta-software.com/link.php?en=353

Bedag Relies on Delta ADS on Visual Studio

A uniform process control and development environment for the different programming languages is the goal of many enterprises. Bedag Informatik AG (Bedag) has mastered this step, it unifies its development processes and thus ensures a more efficient and future-proof application development.

More here:

https://delta-software.com/link.php?en=6291

Aquila Heywood introduces ADS on Eclipse

Aquila Heywood improves its Delta ADS software development by moving to Eclipse and Update to Delta ADS 6.

More here:

https://delta-software.com/link.php?en=6301

B+S Improves Efficiency of Software Development with ADS 6

B+S Banksysteme AG (B+S) has one of the largest application systems that has been developed with Delta tools. In April 2012 B+S started to introduce the pre-release of Delta ADS 6 in order to again increase the efficiency and quality of their software development and maintenance.

More here:

https://delta-software.com/link.php?en=6201

4 'As a Woman in IT' with Dr. Daniela Schilling on the KommtVorZone Podcast

How do you run a company in an industry that consists almost entirely of men?

Als Frau in der IT mit Dr. Daniela Schilling

Dr. Daniela Schilling, CEO of Delta Software Technology, was a guest on Sophia Herrmann's show. She shared her journey from studying computer science to leading a family business, talked about her experiences as a woman in IT and what has shaped her in her role as a leader.

This episode focused on how a woman found her place in a rare yet essential niche within IT while simultaneously taking on a leadership role in a male-dominated field. At the same time, this episode shows that IT has much less to do with the taciturn nerd in a hoodie than Hollywood movies would make us believe.

The discussion also touched on why good leadership - especially in complex fields - requires not only technical expertise but, above all - particularly in IT - communication skills. The focus was on female leadership, role models, the next generation and the question of how job profiles and role models must change so that more women can find their place in technical professions and leadership roles.

“Why are YOU, as a woman, studying computer science?” - a question that still hasn’t gone away.

Her doctoral advisor told her she no longer needed to learn the COBOL language because it was becoming obsolete. Today, she makes her living with it.

She ignored the male voices and forged her own path. Today, many years later, she has long been firmly established as a member of the executive board at an IT company that manages an extremely sensitive infrastructure essential to our daily lives. “If you can’t see my work, then everything’s fine!”

An episode about modern leadership, women’s presence in a male-dominated field, and the courage to take on responsibility in your own way.

It’s worth a listen - even for men!

Order Number: NL21026.02 – July 2026

Delta Software Technology GmbH · Eichenweg 16 · 57392 Schmallenberg, Germany · Phone: +49 2972 9719-0 · info@delta-software.com · www.delta-software.com