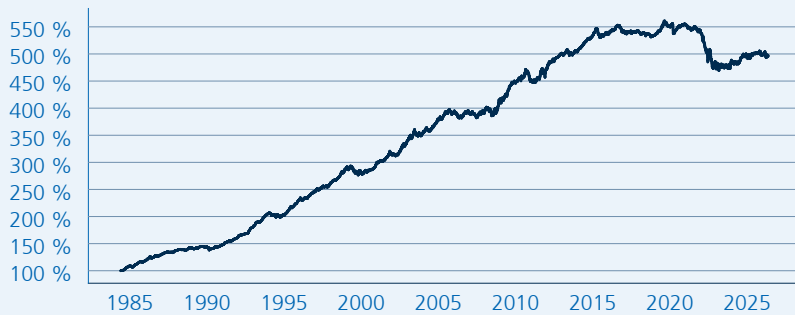




## WERTENTWICKLUNG IN DER VERGANGENHEIT\*

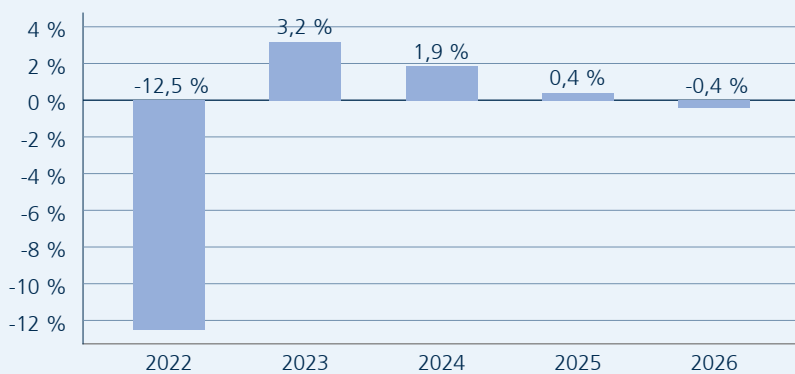
### Indexierte Wertentwicklung



## WERTENTWICKLUNG ÜBER VERSCHIEDENE ZEITRÄUME

1 Monat	3 Monate	6 Monate	lfd. Jahr	1 Jahr
0,17 %	-1,35 %	-0,70 %	-0,43 %	-0,30 %
3 Jahre p.a.	5 Jahre p.a.	10 Jahre p.a.	Seit Auflage p.a.	
1,08 %	-1,84 %	-0,93 %	3,89 %	
3 Jahre	5 Jahre	10 Jahre	Seit Auflage	
3,27 %	-8,89 %	-8,90 %	395,90 %	

## WERTENTWICKLUNG PRO KALENDERJAHR



## ROLLIERENDE 12-MONATS ENTWICKLUNG

13.05.22-13.05.23	13.05.23-13.05.24	13.05.24-13.05.25	13.05.25-13.05.26
-5,38 %	0,73 %	2,83 %	-0,30 %

\*Historische Wertentwicklungen sind keine Garantie für eine ähnliche Entwicklung in der Zukunft. Diese ist nicht prognostizierbar. Die Berechnung der Wertentwicklung erfolgt gemäß BVI-Methode.

## 13. MAI 2026

### Anlageziel

Das Fondsvermögen wird überwiegend in auf Euro lautende verzinsliche Wertpapiere weltweiter Emittenten angelegt. Darunter fallen beispielsweise Staatsanleihen, Pfandbriefe als auch Unternehmensanleihen. In Anleihen, die von der Bundesrepublik Deutschland begeben oder garantiert werden, dürfen mehr als 35 Prozent des Fondsvermögens investiert werden. Die Anlagestrategie orientiert sich an einem Vergleichsmaßstab, wobei versucht wird, dessen Wertentwicklung zu übertreffen. Das Fondsmanagement kann durch aktive Über- und Untergewichtung einzelner Vermögenswerte wesentlich – sowohl positiv als auch negativ – von diesem Vergleichsmaßstab abweichen. Darüber hinaus sind Investitionen in Titel, die nicht Bestandteil des Vergleichsmaßstabs sind, jederzeit möglich.

```

class MirrorX(object):
    """This adds an
    """
    def __init__(self, modifier_ob):
        self.mirror_mod = modifier_ob.modifiers.n
        self.mirror_ob = modifier_ob

    def set_mirror_object(self, mirror_ob):
        self.mirror_mod.mirror_object = mirror_ob

    def __str__(self):
        if_operation == "MIRROR_X":
            mirror_mod.use_x = True
            mirror_mod.use_y = False
            mirror_mod.use_z = False
        elif_operation == "MIRROR_Y":
            mirror_mod.use_x = False
            mirror_mod.use_y = True
            mirror_mod.use_z = False
        elif_operation == "MIRROR_Z":
            mirror_mod.use_x = False
            mirror_mod.use_y = False
            mirror_mod.use_z = True

    def __repr__(self):
        return f'MirrorX(mirror_mod={self.mirror_mod}, mirror_ob={self.mirror_ob})'

    def __getitem__(self, item):
        return self.mirror_mod[item]

    def __setitem__(self, item, value):
        self.mirror_mod[item] = value

    def __delitem__(self, item):
        del self.mirror_mod[item]

    def __len__(self):
        return len(self.mirror_mod)

    def __iter__(self):
        return iter(self.mirror_mod)

    def __contains__(self, item):
        return item in self.mirror_mod

    def __add__(self, other):
        return self.mirror_mod + other

    def __sub__(self, other):
        return self.mirror_mod - other

    def __mul__(self, other):
        return self.mirror_mod * other

    def __div__(self, other):
        return self.mirror_mod / other

    def __mod__(self, other):
        return self.mirror_mod % other

    def __pow__(self, other):
        return self.mirror_mod ** other

    def __float__(self):
        return float(self.mirror_mod)

    def __int__(self):
        return int(self.mirror_mod)

    def __bool__(self):
        return bool(self.mirror_mod)

    def __hash__(self):
        return hash(self.mirror_mod)

    def __eq__(self, other):
        return self.mirror_mod == other

    def __ne__(self, other):
        return self.mirror_mod != other

    def __lt__(self, other):
        return self.mirror_mod < other

    def __le__(self, other):
        return self.mirror_mod <= other

    def __gt__(self, other):
        return self.mirror_mod > other

    def __ge__(self, other):
        return self.mirror_mod >= other

    def __getitem__(self, item):
        return self.mirror_mod[item]

    def __setitem__(self, item, value):
        self.mirror_mod[item] = value

    def __delitem__(self, item):
        del self.mirror_mod[item]

    def __len__(self):
        return len(self.mirror_mod)

    def __iter__(self):
        return iter(self.mirror_mod)

    def __contains__(self, item):
        return item in self.mirror_mod

    def __add__(self, other):
        return self.mirror_mod + other

    def __sub__(self, other):
        return self.mirror_mod - other

    def __mul__(self, other):
        return self.mirror_mod * other

    def __div__(self, other):
        return self.mirror_mod / other

    def __mod__(self, other):
        return self.mirror_mod % other

    def __pow__(self, other):
        return self.mirror_mod ** other

    def __float__(self):
        return float(self.mirror_mod)

    def __int__(self):
        return int(self.mirror_mod)

    def __bool__(self):
        return bool(self.mirror_mod)

    def __hash__(self):
        return hash(self.mirror_mod)

    def __eq__(self, other):
        return self.mirror_mod == other

    def __ne__(self, other):
        return self.mirror_mod != other

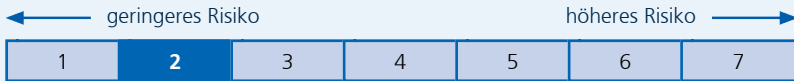
    def __lt__(self, other):
        return self.mirror_mod < other

    def __le__(self, other):
        return self.mirror_mod <= other

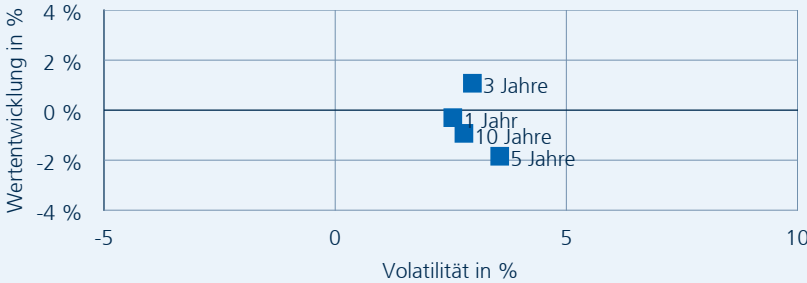
    def __gt__(self, other):
        return self.mirror_mod > other

    def __ge__(self, other):
        return self.mirror_mod >= other
    
```

**RISIKOINDIKATOR\***



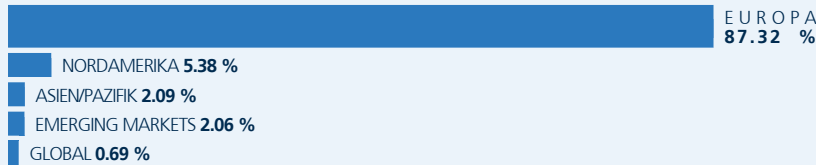
**Risiko-Rendite-Diagramm**



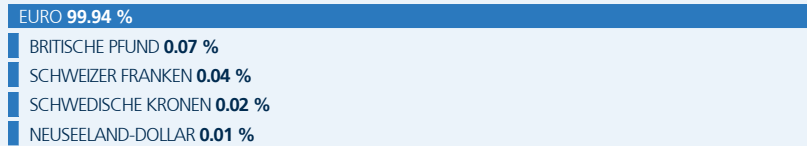
\*SRI gemäß Basisinformationsblatt

**PORTFOLIOSTRUKTUR**

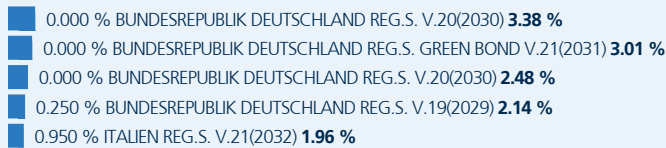
**Top 5 Länderverteilung**



**Top 5 Währungsverteilung**



**Top 5 Holdings**



**Rechtliche Hinweise**

Datenquelle für Fondsinformationen: cleversoft GmbH. Die von der cleversoft GmbH bereitgestellten Informationen stellen keine Empfehlungen dar und dienen nicht der Anlageberatung. Insbesondere stellen diese Informationen keine Finanzanalyse im Sinne von § 34 b WpHG dar. Die cleversoft GmbH trifft keinerlei Aussage zur bisherigen und zukünftigen Wertentwicklung der angezeigten Wertpapiere. Wegen der Dynamik der Finanzmärkte muss jegliche Haftung im Zusammenhang mit der Nutzung der Informationen oder dem (evtl. auch irrtümlichen) Vertrauen auf deren Richtigkeit, Vollständigkeit oder Aktualität ausgeschlossen werden. Informieren Sie sich daher vor dem Fondserwerb auf der Internetseite der jeweiligen Fondsgesellschaft.

**Fonds-Fakten**

Fondskategorie	Rentenfonds
WKN	849106
ISIN	DE0008491069
Auflagedatum	01. Juni 1984
Fondsvolumen	5,75 Mrd. EUR (13.05.2026)
Kapitalverwaltungs-gesellschaft	Union Investment Privatfonds GmbH
Fondsmanagement	Union Investment Gruppe
Depotbank	DZ Bank AG Deutsche Zentral-Genossenschaftsbank
Sitzland	Deutschland
Fondswährung	EUR
Rücknahmepreis	59,77 EUR
Ertragsverwendung	Ausschüttend
Geschäftsjahr	01. Oktober - 30. September
Transparenzverord-nung (EU) 2019/2088	-

**Fonds-Konditionen**

Ausgabeauschlag	Ein Ausgabeaufschlag fällt im Rahmen der fondsge-bundenen Rentenversiche-rungen von R+V nicht an.
Max. Verwaltungs-vergütung p. a.	0,80 %
Max. Fondsmanage-ment Gebühr p.a.	0,80 %
Max. Depotbankver-gütung p.a.	0,00 %
Laufende Kosten p.a.	0,70 % (16.04.2026)
Erfolgsabhängige Vergütung	0,00 %

**Kennzahlen**

Volatilität		Maximum Drawdown	
1 Jahr	+2,54 %	1 Jahr	-2,49 %
3 Jahre	+2,97 %	3 Jahre	-2,49 %
5 Jahre	+3,56 %	5 Jahre	-14,79 %

Sharpe Ratio		Verlustdauer in Monaten	
1 Jahr	-0,89	1 Jahr	2
3 Jahre	-0,60	3 Jahre	2
5 Jahre	-1,02	5 Jahre	7

Informationen erhalten Sie in den Volksbanken und Raiffeisenbanken, R+V-Agenturen sowie bei der Direktion der Gesellschaften der R+V Versicherungsgruppe, Raiffeisenplatz 1, 65189 Wiesbaden.

Telefon: 0800 533-1171

Kostenfrei aus allen deutschen Fest- und Mobilfunknetzen.

[www.ruv.de](http://www.ruv.de)

R+V Lebensversicherung AG